Modeling curiosity for intelligent agents

Wu, Qiong

2014

Wu, Q. (2014). Modeling curiosity for intelligent agents. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62194>

<https://doi.org/10.32657/10356/62194>

MODELING CURIOSITY FOR INTELLIGENT AGENTS

A thesis submitted to
School of Computer Engineering
Nanyang Technological University

by

Qiong Wu

in fulfilment of the requirements
for the degree of Doctor of Philosophy

August 2014

Acknowledgement

First and foremost, I want to thank my supervisor, Dr. Chunyan Miao, for her constant support and encouragement in all aspects of my research, career, and even personal issues. Dr. Miao has been not only a great advisor but also a cordial friend and a personal mentor to me. She gave me a lot of freedom in pursuing my broad interests and taught me how to become a researcher. I thank her for offering me this valuable opportunity to learn and grow. This thesis would not have been conceptualized without her kind help.

I am especially grateful to Dr. Zhiqi Shen and Dr. Bo An for their helpful and insightful comments. Special thanks go to Dr. An for his guidance and encouragement on the curious learning companion problem. I also want to thank my friend and colleague, Dr. Siyuan Liu and Yuan Liu, for their discussion and sharing of knowledge in the curious recommender agent problem. I want to express special thanks to my cherished friend Kartick for all his kind help and his great friendship. Besides, I want to thank all my colleges who are working at the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, for providing me with such a warm and open environment to conduct my work.

Finally, I want to express my deepest love to my family. To my parents, I can never express enough how much I love them and how much I thank for the endless love and support they give me. Very special thanks go to my lifetime partner, Yue, for his deep love, understanding, support, commitment and all the great happiness he brought me. This thesis is dedicated to them.

Contents

Acknowledgements	i
Abstract	v
List of Figures	viii
List of Tables	x
List of Symbols	xii
I Introduction	1
1 Introduction	2
1.1 Background and Motivation	2
1.2 Research Issues and Challenges	5
1.3 Major Contributions	8
1.4 Thesis Organization	10
2 Literature Review	12
2.1 Curiosity in Psychology	12
2.1.1 Underpinnings of Curiosity	13
2.1.2 Arousal Mechanisms of Curiosity	16
2.2 Computational Curiosity in AI	19
2.2.1 Computational Models of Curiosity	20
2.2.2 Major Applications	31
2.3 Summary	36

II	Computational Framework of Curiosity	38
3	Computational Model of Curiosity	39
3.1	Standard Agent	40
3.1.1	Environment	40
3.1.2	Internal Functions	41
3.2	Memory and Learning	43
3.2.1	Memory	44
3.2.2	Learning	44
3.3	Curious Functions	46
3.3.1	Stimulus Detection	47
3.3.2	Interest Evaluation	47
3.3.3	Meta-level Decision-making	48
3.4	Summary	50
4	Curiosity-driven Extreme Learning Machine	51
4.1	Introduction	51
4.2	Curious Extreme Learning Machine Classifier (C-ELM)	53
4.2.1	The Internal Cognitive Component: SLFN	54
4.2.2	Curious Functions	55
4.3	Performance Evaluation of C-ELM	60
4.3.1	Performance Measures	60
4.3.2	Performance Study on Multi-category Classification Problems	61
4.3.3	Performance Study on Binary Classification Problems	62
4.4	Summary	63
5	Curiosity-driven Recommendation	65
5.1	Introduction	65
5.2	Curious Recommender Agent	66
5.2.1	Environment	67
5.2.2	The Internal Function: Rating Prediction	68
5.2.3	Curious Functions: Stimulus Detection and Interest Evaluation	69

5.3	Performance Evaluation of the Curious Recommender Agent	74
5.3.1	Datasets	74
5.3.2	Metrics	75
5.3.3	Methods	76
5.3.4	Experiment Results	77
5.4	Summary	80
III	Application in Virtual Worlds for Education	81
6	Curious Peer Learner	82
6.1	Introduction	82
6.2	Curious Peer Learner	84
6.2.1	Virtual Learning Environment	84
6.2.2	Internal Functions	86
6.2.3	Curious Functions: Stimulus Detection and Interest Evaluation	89
6.3	Performance Evaluation of the Curious Peer Learner	97
6.3.1	Virtual Singapura	97
6.3.2	Experiment Setup	99
6.3.3	Experiment Results	102
6.4	Summary	107
7	Curious Learning Companion	109
7.1	Introduction	109
7.2	Curious Learning Companion	110
7.2.1	Virtual Learning Environment	111
7.2.2	Internal Functions: Planning and Learning	113
7.2.3	The Curious Function: Stimulus Detection	114
7.3	Performance Evaluation of the Curious Learning Companion	117
7.3.1	Implementation in Virtual Singapura	117
7.3.2	Hypotheses	119
7.3.3	Field Study	120

7.3.4	Experiment Results	121
7.3.5	Discussion	123
7.4	Summary	124
IV	Conclusion	128
8	Conclusion and Future Directions	129
8.1	Thesis Summary	129
8.1.1	Computational Framework of curiosity	129
8.1.2	Application in Virtual Worlds for Education	131
8.2	Future Directions	132
8.2.1	Modeling Aspects of Computational Curiosity	133
8.2.2	Research Areas that Computational Curiosity will Impact	135
	List of Publications	139
	References	141
	Appendices	141

Abstract

In recent years, researchers have shown an increasing interest in studying various characteristics of intelligent agents, such as emotion, negotiation, trust, and persuasion. However, agents with these characteristics still lack the capability to direct their attention towards novelty and seek interestingness. In human beings, such capability allows us to achieve individual growth and a closer interpersonal relationship, which is governed by curiosity, the critical motivation associated with exploration, learning, and interest. Therefore, it can be envisioned that a computational model of curiosity may enable a learning agent to achieve better learning efficiency, a recommender agent to predict interesting items, or a pedagogical agent to demonstrate believable learning behaviors. These thoughts inspire us to model human-like curiosity for intelligent agents and study the impact brought by this novel characteristic.

A computational framework of curiosity is developed in this thesis, which is largely inspired from the literature in psychology. Berlyne's and Wundt's theories on curiosity are chosen to be the basis for conducting this work due to their long lasting impact in the psychology of curiosity. Berlyne's theory identifies several key factors that govern the process of curiosity arousal, such as novelty, surprise, uncertainty, conflict, and complexity. Wundt's theory postulates an inverted U-shape relationship between the stimulation level and the hedonic value, which states that only optimal stimulation level results in curiosity. Based on these theories, we propose a computational framework of curiosity for intelligent agents. This framework is composed of a generic computational model of curiosity, a curiosity-driven learning algorithm, and a curiosity-driven recommendation algorithm. The computational model of curiosity consists of abstract functions and their interactions between each other. Representing computational models for intelligent agents with abstract functions makes them general enough to allow different implementations in different application contexts. The curiosity-driven learning

algorithm realizes the computational model of curiosity in a fast neural learning agent: the extreme learning machine. Performance comparisons with other popular learning algorithms in the literature on benchmark classification problems demonstrate a superior learning and generalization ability of the curiosity-driven learning algorithm. The curiosity-driven recommendation algorithm realizes the computational model of curiosity in a social recommender agent. Experimental studies with large scale real world data sets show that curiosity significantly enhances the agent’s recommendation diversity and coverage, while maintaining a sufficient level of accuracy.

To evaluate the practical values of the proposed computational model of curiosity, we study it in the domain of virtual worlds for educational purposes; as it has been shown in the literature of education that curiosity is an important driving force for child development, scientific research, and educational achievements. Two types of pedagogical agents are chosen for study, a virtual peer learner and a virtual learning companion. The virtual peer learner is a Non-player Character that aims to provide a believable virtual environment without direct interactions with users, whereas the virtual learning companion directly interacts with users to enhance their learning experience. Studies on a curious peer learner through computer simulations show that curiosity enables a virtual peer learner to demonstrate human-like learning behaviors, which significantly enhances believability. Studies on a curious learning companion with human users show that curiosity of the virtual learning companion significantly improves the users’ learning experience from several aspects, including learning gains, self-efficacy, and interest.

List of Figures

1.1	An overview of the thesis work.	4
2.1	The Wundt curve (adopted from Berlyne [8]).	19
2.2	A general appraisal process of computational curiosity.	20
3.1	The architecture of a standard agent.	43
3.2	The architecture of an agent with memory.	44
3.3	The architecture of a learning agent.	46
3.4	The architecture of a curious agent.	49
5.1	The environment of the curious recommender agent.	67
5.2	Performance Results on Coverage, Diversity, MAE and RMSE with Douban and Flixster Datasets	78
6.1	An illustration of the interactions between the curious peer learner and the VLE.	84
6.2	The concept map for the learning task ‘photosynthesis’	86
6.3	The Wundt curve	94
6.4	The architecture of the curious peer learner	95
6.5	Screen shots of the virtual environment in VS	98
6.6	An illustration of the interaction scenarios for a curious peer learner in VS	98
6.7	The curious peer learner’s learning behavior in training and testing phases	103
6.8	The curious peer learner’s learning behavior in the testing phase	105
6.9	The non-curious peer learner’s learning behavior in the testing phase . .	105
7.1	An illustration of the role of a virtual learning companion in VLE.	110

7.2	The partial plan structure for the goal of “driving to the supermarket”. .	114
7.3	An illustration of the virtual companion’s curious questions: “What does osmosis transport? Let’s ask the Sage.”	118
7.4	The architecture of the curious learning companion.	119
7.5	The mean scores of control and experiment group for seven aspects: learning gains, facilitating learning, credible, human-like, engaging, self-efficacy, and interest.	124

List of Tables

2.1	A taxonomy of the existing computational models of curiosity in literature	22
4.1	The specification of benchmark data sets on classification problems . . .	61
4.2	Performance comparison between C-ELM and other popular classifiers on the multi-category classification problems	62
4.3	Performance comparison between C-ELM and other popular classifiers on the binary classification problems	63
5.1	Statistics of User-Item Matrix of Douban.	74
5.2	Statistics of Friend Network of Douban.	74
5.3	Statistics of User-Item Matrix of Flixster.	75
5.4	Statistics of Friend Network of Flixster.	75
6.1	The parameter setting for the curious peer learner and the non-curious peer learner.	100
6.2	The specification of function mapping for the curious peer learner. . . .	102
6.3	The comparison of exploration breath and exploration depth between the virtual peer learners with curiosity and without curiosity.	106
7.1	The exemplary question templates for the curious learning companion. .	116
7.2	The performance comparison between the basic learning companion and the curious learning companion on learning gains	127
7.3	The performance comparison between the basic learning companion and the curious learning companion on agent persona	127
7.4	The performance comparison between the basic learning companion and the curious learning companion on self-efficacy	127

7.5	The performance comparison between the basic learning companion and the curious learning companion on interest	127
8.1	An illustration of the interest deprivation (I/D) theory of curiosity	134

List of Symbols

AGT	The agent
E	The environment behavior function
P	The perceiving function
D	The decision-making function
A	The acting function
L	The learning function
M	The memory update function
S	The stimulus detection function
I	The interest evaluation function
<i>E</i>	The environment
<i>S</i>	The set of environmental state
<i>A</i>	The set of agent actions
<i>A*</i>	The set of agent's previous actions
<i>P</i>	The set of agent percepts
<i>P*</i>	The set of agent's previous percepts
<i>D</i>	The set of agent decision points
<i>D*</i>	The set of agent's previous decision points
<i>M</i>	The agent memory
<i>L</i>	The set of agent learning rules
<i>V</i>	The set of collative variables
<i>Em</i>	The set of curiosity-related emotions
<i>s</i>	An environmental state
<i>a</i>	An agent action
<i>p</i>	An agent percept

d		An agent decision point
l		An agent learning rule
$\mathcal{N}$		Novelty
$\mathcal{S}$		Surprise
$\mathcal{U}$		Uncertainty
$\mathcal{F}$		Conflict
$\mathcal{H}$		Change
$\mathcal{P}$		Complexity
$\mathcal{A}$		Anxiety
$\mathcal{C}$		Curiosity
$\mathcal{B}$		Boredom

Part I

Introduction

Chapter 1

Introduction

1.1 Background and Motivation

In recent years, researchers have shown an increasing interest in studying intelligent agents with various characteristics, such as affective agents [93] [103] [143], negotiation agents [43], trust agents [95], persuasive agents [1], and pedagogical agents [45]. Each of the characteristics brings a new capability to intelligent agents and enhances certain aspect of their performances. These agents, however, still lack the capability to direct their attention towards novelty or to seek interestingness. In human beings, this capability is commonly observed, which is driven by the natural motivation: curiosity [52]. Curiosity has been consistently recognized as the critical motivation that is associated with exploratory behaviors such as exploration, investigation, and learning [51]. It has been identified as a driving force for child development, scientific research, and educational achievements [65]. According to Kashdan [49], curiosity benefits human beings at two levels: the individual level and the social level. At the individual level, curiosity is associated with individual growth, as an “innate love of learning without the lure of any profit” [65]. At the social level, curiosity is an ingredient for enhancing interpersonal relationships, through infusing energy and passion into social interactions [52].

The advantages of curiosity have inspired researchers to devise computational forms of curiosity to endow artificial beings (or agents) with desirable functions. For example, a curious design agent can arrange art exhibits to elicit the curiosity of their viewers and provide an aesthetically pleasing experience [108]; a curious exploratory agent has been shown to achieve higher learning efficiency in unknown environments [73]. From

a machine learning perspective, curiosity has been proposed as algorithmic principles to focus learning on novel and learnable regularities in contrast to irregular noises [111, 87, 90, 48]. These algorithm principles make the agents to be “explorative” and allow them to have “the desire to improve the model’s knowledge about the world” [113, 116], which have shown success in speeding up learning and building unsupervised developmental robotics [116, 87].

The existing models of curiosity, however, are mostly built heuristically, which may not capture the essence of the underlying mechanism of human curiosity. Moreover, it is difficult to transform these models to other domains of application. For example, the algorithmic principles that focus a machine learning agent on novel and learnable regularities cannot efficiently guide a recommender agent to detect surprising rating information on social networks. They also may not guide a virtual learning companion to self-direct to knowledge gaps and demonstrate human-like learning behaviors in a virtual learning environment. Hence, it is necessary to build a generic computational model of curiosity for intelligent agents, which can be implemented across domains to support curious behaviors for agents with different goals.

In order to build a generic model of curiosity for intelligent agents, we base our study on psychological theories. Berlyne [8], who believed that curiosity can be externally stimulated, identified the key factors that determine the stimulation level of external stimuli, including novelty, surprise, uncertainty, conflict, and complexity. However, a higher level of stimulation may not always lead to a higher level of curiosity. Wundt [149] postulated an inverted U-shape relationship between the overall stimulation level of curiosity and emotion states, where too little stimulation results in boredom, too much stimulation results in anxiety and only optimal stimulation results in curiosity. Based on these theories, we propose a generic computational model of curiosity for intelligent agents. This computational model of curiosity consists of abstract functions and their interactions between each other. Representing computational models for intelligent agents with abstract functions makes them general enough to allow different implementations in different application contexts [142].

Based on the proposed computational model of curiosity, we develop a curiosity-driven learning algorithm and a curiosity-driven recommendation algorithm. The curiosity-driven learning algorithm showcases how curiosity benefits an agent at the individual

Figure 1.1: An overview of the thesis work.

development level, whereas the curiosity-driven recommendation algorithm demonstrates how curiosity benefits an agent at the social recommendation level. All the three components combine to form a computational framework of curiosity. The curiosity-driven learning algorithm realizes the generic computational model of curiosity in a fast neural learning agent: the extreme learning machine [38], and is referred to as the Curious Extreme Learning Machine (C-ELM). Experimental comparisons with other popular classifiers on benchmark classification problems show a superior learning and generalization performance of C-ELM. In the context of social recommendation, we develop a curiosity-driven recommendation algorithm by realizing the generic computational model of curiosity in a recommender agent. Experimental studies with large scale real world

data sets show that curiosity significantly enhances the recommendation coverage and diversity, while maintaining a sufficient level of accuracy.

To evaluate the practical values of the proposed computational model of curiosity, we study it in the domain of virtual worlds for educational purposes. It has been shown in educational studies that curiosity is an important driving force for child development, scientific research, and educational achievements [65]. Hence, it can be envisioned that considering the concept of curiosity in the design of virtual worlds may enrich their educational potentials. In this thesis, two types of pedagogical agents are chosen for study in virtual world based learning environments, a virtual peer learner and a virtual learning companion. The virtual peer learner is a Non-player Character that aims to provide a believable virtual environment without direct interactions with users, whereas the virtual learning companion directly interacts with users to enhance their learning experience. It can be expected that curiosity may allow the virtual peer learner to demonstrate more human-like learning behaviors in a virtual learning environment, and adds new ingredients into the interactions provided by the virtual learning companion.

Based on the above introduction, an overview of the thesis work is summarized in Figure 1.1. Next, we will discuss the key research issues and challenges tackled in this thesis.

1.2 Research Issues and Challenges

Major research issues and challenges tackled in this thesis include:

- **Generic model of curiosity:** Although curiosity has attracted a lot of interest from researchers in the domain of artificial intelligence, existing computational models of curiosity are often built heuristically. These models treat curiosity as algorithm principles to enhance machine learning ability [111, 87, 90, 48], or as novelty detection principles to find interesting design patterns in the field of artificial creativity [108]. However, these models are domain specific and cannot be generally applied for agents with different goals (other than learning or detecting novel design patterns) to demonstrate curiosity. For example, the algorithm principles that guide a machine learning agent to learn surprising information cannot efficiently

guide a learning companion agent to demonstrate complex curious behaviors in a human-agent interaction context. Hence, a big challenge for researchers is to build a generic computational model of curiosity that can be applied for intelligent agents with different goals to demonstrate human-like curious behaviors. In order to simulate a human-like curiosity appraisal process, this generic computational model of curiosity shall not be built heuristically, but should be rooted in the psychology of curiosity.

- **Learning agent:** As discussed above, being an important motivation for learning, curiosity has been modeled for various types of learning agents, e.g., reinforcement learning agents [4] and neural network based learning agents [111], to enhance their learning and generalization performance. However, these models of curiosity are often heuristically built and be defined as simple algorithm principles [145], which diverge far from the mechanism of human curiosity and may not capture the essence of curiosity for learning completely. Hence, there is a challenge to model human-like curiosity that is transposed from theories in psychology for learning agents and study the impact of such a model of curiosity on their performances. In this thesis, we take a fast neural learning algorithm referred to as the Extreme Learning Machine (ELM) as an example and study the influence of psychological inspired computational curiosity on its performance.
- **Recommender agent:** The ultimate goal of a recommender system is to suggest particularly interesting items, in addition to indicating those that should be filtered out [99]. Traditional recommender systems are built based on a general consensus that user preferences reflect their underlying interests. Hence, various collaborative filtering techniques [104] have been proposed to discover items that best match users' preferences. However, determining interestingness based on user preferences alone is far from sufficient. According to the psychology of interest [125], the appraisal of interestingness in human beings is closely related to curiosity. Instead of focusing on a person's preferences, curiosity highlights the impact of novelty and surprisingness on a person's feeling of interest. Therefore, in this thesis, we take a new angle to study the interestingness of recommendation and introduce a novel

dimension of social information into the traditional recommender systems: social curiosity.

- **Virtual peer learner:** In virtual world based learning environments, virtual peer learners are important ingredients to form a social learning environment for users. A key design aspect of the virtual peer learners is their believability, which is critical to achieve suspension of disbelief in a virtual world. In order to design a virtual peer learner that can emulate a real student and behave naturally in virtual learning environments, a psychologically inspired approach is necessary. Studies in psychology have shown that curiosity is the motivation that links cues reflecting novelty and challenge with natural behaviors such as exploration, investigation and learning [52]. In Reiss's 16 basic desires that motivate our actions and shape our personalities [97], curiosity is defined as "the need to learn". Hence, as a natural motivation in human beings to drive their learning behaviors, modeling curiosity in virtual peer learners may lead them to demonstrate human-like behaviors in a learning environment. In this thesis, we study the believability of virtual peer learners through simulating human-like curiosity in them.
- **Virtual learning companion:** Virtual peer learners are background characters to enrich the virtual learning environment, which do not interact with users. However, many educational practices are instantiated through the interactions between teachers and students or between peer learners [66]. Hence, a virtual learning companion that can directly interact with the users would be interesting to enhance their learning experiences. Curiosity is an emotional motivation related to exploratory behaviors such as learning, investigation, and exploration, which drives human beings to ask questions and explore for answers [65]. Hence, modeling human-like curiosity in a virtual learning companion may allow the agent to discover knowledge gaps and ask relevant questions. These questions can add new ingredients into the interactions provided by the companion, which may help human learners notice the weakness in their knowledge structure and motivate them to actively explore the virtual learning environment. Through the design and development of a curious learning companion, we study the influence of an agent's curiosity on students' learning experiences in the virtual learning environment.

Next, we will discuss the major contributions of this thesis.

1.3 Major Contributions

Major contributions of this thesis include:

- **Computational framework of curiosity:** A computational framework of curiosity is proposed for intelligent agents, which consists of a generic computational model of curiosity, a curiosity-driven learning algorithm, and a curiosity-driven recommendation algorithm.
 - **Computational model of curiosity:** A computational model of curiosity for intelligent agents is proposed, which consists of abstract functions and their interactions between each other. Representing agent models with abstract functions makes them general enough to allow different implementations in different application contexts [142]. The curious functions are transposed based on Berlyne’s and Wundt’s theories in human psychology, which ensures the fidelity of our model.
 - **Curiosity-driven learning algorithm:** A curiosity-driven learning algorithm referred to as the curious extreme learning machine classifier (C-ELM) is realized based on the proposed computational model of curiosity. C-ELM treats each input data as a curious stimulus and performs curiosity appraisal towards each input data based on four collative variables: novelty, uncertainty, conflict, and surprise. Three learning strategies can be chosen from based on the curiosity appraisal results, including neuron addition, neuron deletion, and parameter update. C-ELM enhances traditional ELM algorithms with the evolving capability, which determines optimal network structure dynamically based on training data. Also, C-ELM reduces partially the random effect of traditional ELM algorithms by selecting RBF centers based on data instead of random assignment. Moreover, C-ELM employs a novel neuron deletion strategy which is based on conflict resolution. Empirical studies of C-ELM show that the proposed approach leads to compact network structures and achieves

better generalization performance with fast response speed, comparing with the traditional ELM and other popular classifiers.

- **Curiosity-driven recommendation algorithm:** A curiosity-driven recommendation algorithm is realized based on the proposed computational model of curiosity. The computational model is implemented based on the popular predictive model by matrix factorization, due to its high recommendation accuracy and efficiency in dealing with large scale user-item rating matrices [144]. A general and parameter-free model is proposed for measuring user curiosity in the social contexts. This model takes into consideration the different responses given by a user to different friends' surprising ratings. Three strategies are also proposed and compared for evaluating user curiosity when multiple friends give surprising ratings to the same item. The curiosity-driven recommendation algorithm is carefully studied with two large scale real world data sets. The experimental results show that social curiosity significantly improves recommendation diversity and coverage, while maintaining a sufficient level of accuracy.
- **Application in virtual worlds for education:** In virtual world based learning environment, the proposed computational framework of curiosity is carefully studied with two types of pedagogical agents: a curious peer learner and a curious learning companion.
 - **Curious peer learner:** A curious peer learner is realized based on the proposed computational model of curiosity. The computational model is implemented based on the computational theories of concept maps and reinforcement learning. Concept maps is a semantic knowledge representation method that stems from the learning theory of constructivism [83], which allows the knowledge representation to be easily interpreted by both the agents and the human students. The emotion states derived from the curious functions in the generic computational model serve as intrinsic reward functions to regulate the agent's reinforcement learning process. The experimental study through computer simulations shows that curiosity leads the virtual peer learners to

behave naturally in a virtual learning environment as real human students do, which significantly enhances the believability of the virtual peer learners.

- **Curious learning companion:** A curious learning companion is realized based on the proposed computational model of curiosity. The computational model is implemented using plan-based knowledge representations and employs decision theoretic planning techniques. Plan-based approaches now resides in the core of a number of complex real-time autonomous systems, due to the effectiveness of explicit representations of plans, which enables the easy reasoning of future outcomes [28]. The curious learning companion is carefully studied with human users. Experimental results show that a curious learning companion significantly enhances human learners’ learning experiences from several aspects, including learning gains, self-efficacy, and interest. Also, the results supported that human learners perceive a curious learning companion’s persona more favorably.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 provides a comprehensive literature review on curiosity from two fields: psychology and computation. Knowledge from the field of psychology provides insights into modeling curiosity computationally. The state-of-the-art literature on computational curiosity reveals future directions for research.

In *Chapter 3*, a generic computational model of curiosity is proposed for intelligent agents. This computational model of curiosity serves as a basis for the different realizations in various types of intelligent agents presented in the subsequent chapters.

Chapter 4 presents a curiosity-driven learning algorithm. This algorithm realizes the generic computational model of curiosity in an extreme learning machine based classifier, which is referred to as the Curious Extreme Learning Machine classifier (C-ELM). The performance of C-ELM is measured against other popular classifiers in the literature on benchmark data sets.

Chapter 5 presents a curiosity-driven recommendation algorithm. This algorithm realizes the generic computational model of curiosity in the popular matrix factorization

algorithm for recommendation problems. The curiosity-driven recommendation algorithm is able to recommend items that both match the target users' preferences and are surprising, so as to stimulate their curiosity. The performance of the curiosity-driven recommendation algorithm is evaluated by two large scale real world data sets.

In *Chapter 6*, a curious peer learner is developed. It is a non-player character that aims to provide a believable virtual learning environment for users. The effects brought by curiosity to virtual peer learners is studied through computer simulations.

In *Chapter 7*, a curious learning companion is developed. It aims to enhance users' learning experience through providing meaningful interactions with them. The curious learning companion realizes the generic computational model of curiosity and is carefully evaluated with human users through field studies.

Chapter 8 concludes the thesis by clarifying the main contributions and providing suggestions for future research.

Chapter 2

Literature Review

This chapter provides a comprehensive review on curiosity from two fields: psychology and computation. Knowledge from the field of psychology provides insights into modeling human-like curiosity computationally and form the basis for our research. The state-of-the-art literature on computational curiosity reveals issues and challenges for further studies, which produce the motivations for our research. In this literature review, we start with the theories of curiosity in psychology, from which we derive a two-step process model of curiosity arousal. This two-step process model serves as the basis for analyzing the computational forms of curiosity uniformly. After that, we switch our angle from psychology to computation and review the existing computational models of curiosity and their major applications. The review presented in this chapter has been published in [145]. Next, we will start our review from the field of psychology.

2.1 Curiosity in Psychology

Research on curiosity has attracted psychologists' interest since back in the 1960s, when researchers attempted to uncover the underpinnings of curiosity [65]. The effort continues to date with researchers focusing on interpreting relationships between curiosity and human well-beings [49]. In this section, we review psychological studies on curiosity while answering two questions:

1. what are the underpinnings of curiosity?
2. what is the arousal mechanism of curiosity?

2.1.1 Underpinnings of Curiosity

In human beings, curiosity is closely related to cognition, emotion and behavior. It underlines human cognitive development, aesthetic appreciation, and interpersonal relationships. In this section, we review the literature in psychology on the underpinnings of curiosity and the benefits of curiosity for human beings.

2.1.1.1 Categories of Curiosity

Most psychologists believe that curiosity is an intrinsic motivation driving the cognitive development of both humans and animals alike. Berlyne [8] categorized curiosity along two spectrums: (1) from perceptual curiosity to epistemic curiosity, and (2) from specific curiosity to diversive curiosity. Perceptual curiosity, which resides in the lower level of cognition, stems from the senses of both animals and humans (e.g. senses of touch, vision, taste, etc.). It is defined as “a drive that is aroused by novel stimuli and reduced by continued exposure to these stimuli” [65]. Epistemic curiosity, referred to as “an appetite for knowledge”, is related to the higher level of cognition and believed to be a distinctive human feature. While perceptual curiosity and epistemic curiosity are defined along the lines of “lower” and “higher” levels of cognition, specific curiosity and diversive curiosity are distinguished by the possibility of curiosity having a “direction”. Specific curiosity is aroused by a particular piece of information. Diversive curiosity is a general drive to seek information with no specific direction and is predominantly employed to relieve boredom.

Berlyne’s cognitive account for curiosity has been a theoretical foundation for the majority of recent studies. Litman and Spielberger [64] agreed with Berlyne that there is a salient difference between diversive curiosity and specific curiosity, and conducted experimental analysis to provide scales for measuring both concepts. They further concluded that diversive curiosity and specific curiosity, as well as perceptual curiosity and epistemic curiosity, are “substantially correlated”. Spielberger and Starr [128] associated diversive curiosity with “low-level” stimuli and specific curiosity with “high-level” stimuli. However, Schmitt and Lahroodi [121] disagreed with the notion that curiosity can be diversive. They argued that curiosity can only be specific towards its possessor and objects that cause the curiosity. Instead of diversive curiosity, they referred to the generic desire for knowledge as “inquisitiveness”.

Nevertheless, a general consensus among the psychologists points to the close relationship between curiosity and cognition, as a drive to explore novel stimuli or an appetite for knowledge.

2.1.1.2 Curiosity-Related Emotions

Curiosity is related to emotional constructs. In an early account of curiosity by James [42], curiosity is viewed as an instinctual or emotional response closely related to fear. He believed that curiosity motivates organisms to actively explore their environment, whereas fear tends to turn the organisms away from the risks induced by unfamiliarity. Berlyne's branch of research, referred to as "drive theory" by Loewenstein [65], is based on the assumption that curiosity produces an unpleasant sensation that is reduced by exploratory behaviors. Loewenstein believed that rather than serving a purposive end, the primary objective of satisfying one's curiosity is to induce pleasure.

Wundt [149] introduced the theory of "optimal level of stimulation", which serves as a general rule postulating the relationships between stimulus intensity and the hedonic tone. Based on this theory, Berlyne [8] proposed that there is a need of "intermediate arousal potential" for curiosity to be aroused. Berlyne's theory demonstrates that too little stimulation can result in boredom or inaction, while too much stimulation may result in aversion or withdrawal. Only when the level of stimulation is optimal and pleasurable can exploratory behaviors occur.

From the above discussion, it can be seen that curiosity is closely related to emotional constructs such as "fear", "pleasure", "boredom" and "anxiety". The decision on whether to explore or to avoid a stimulus is driven by emotional comfort and results in behaviors that regulate emotional states.

2.1.1.3 Curiosity-Related Behaviors

The most salient expression of curiosity is through exploratory behaviors, by which curiosity can be satisfied. Berlyne [8] defined two levels of exploratory behaviors, one associated with the perceptual level of curiosity, and the other associated with the epistemic level of curiosity. At each level, the exploratory behaviors can take many forms.

At the perceptual level of curiosity, Berlyne divided exploratory behaviors into three categories according to the nature of responses. He referred to the exploratory behaviors as *orienting responses* if they consist of changes in posture, orientations of sensory organs, or states of sensory organs. The second category of exploratory behaviors is associated with *locomotion*, such as approaching or withdrawing from the stimuli. When an exploratory behavior causes changes in external objects, through manipulation or otherwise, it is called an *investigatory response*.

At the epistemic level of curiosity, Berlyne also defined three categories of exploratory behaviors. The first category is *observation*, which places the subject in contact with external situations that can nourish the pertinent learning process. The second category is *thinking*, which refers to “productive” and “creative” thinking, rather than “reproductive thinking” that only calls up memories to determine how problems should be handled. The last category is *consultation*, which exposes an individual to verbal stimuli issued from other individuals, including questioning and reading, etc.

In summary, through exploratory behaviors, cognitive growth is achieved by “creative thinking” and emotional states are regulated towards a feeling of “pleasure”.

2.1.1.4 Benefits of Curiosity

Current research indicates that curiosity can contribute to human well-beings at two distinct levels: individual level and social level.

At the individual level, curiosity provides an emotional motivation for self development. It is the driving force for child development as well as an important spur for educational achievements [65]. Also, literature in psychology indicates a close relationship between curiosity and aesthetics, humor, and fun. According to Berlyne [8], these behaviors have common motivational factors and are reinforced by common sources of rewards.

At the social level, curiosity can enhance interpersonal relationships. By studying the role of curiosity in conversations, Kashdan et al. [50] suggested that curiosity can build social bonds by promoting behaviors such as engagement, responsiveness, and flexibility. These are desirable behaviors for developing interpersonal relationships and building intimacy. Their findings indicate that curiosity is uniquely related to the development of interpersonal closeness with strangers [51].

2.1.2 Arousal Mechanisms of Curiosity

In this section, we review the literature in psychology on the arousal mechanisms of curiosity. This review will provide insights into possible ways of implementing curiosity in artificial beings and is the basis for conducting our research.

According to Loewenstein [65], existing psychological theories on human curiosity can be divided into three categories: *incongruity theory*, *competence theory*, and *drive theory*. Incongruity theory holds on to the idea that curiosity is evoked by the violation of expectations [31]. Competence theory views curiosity as an intrinsic motivation to master one's environments [138]. Drive theory believes in the existence of a curious drive, either primary (homeostatic generated in a similar way as hunger) or secondary (externally generated by stimuli). As commented by Loewenstein, the incongruity theory and the competence theory suffer from the same deficiency that both fail to offer a comprehensive account of curiosity. Hence, in this work, we focus on the drive theory and adopt Berlyne's interpretation of curiosity.

According to Berlyne [8], traditional psychological research concentrated on problems of *response selection*, which studies what response humans will make to one standard stimulus at a time. However, curiosity deals with a different problem from response selection, which is referred to as *stimulus selection*. Stimulus selection discusses when several conspicuous stimuli are introduced at once, to which stimulus humans will respond. Berlyne has conducted extensive experimental studies to understand the process of stimulus selection and discovered a set of collative variables that govern this process.

2.1.2.1 Collative Variables

Collative variables, according to Berlyne [8], refer to the external factors that govern various forms of stimulus selection. There are four major collative variables, viz., *novelty*, *uncertainty*, *conflict* and *complexity*. Berlyne named these factors collative variables because the evaluation of each variable involves an analysis of similarities and differences between elements in a stimulus pattern. In the following part of this section, the major collative variables are reviewed.

Novelty denotes something new. In accordance with the time when a stimulus has been experienced by an organism, novelty can be divided into *short-term*, *long-term*, and

complete novelty. A stimulus can either be completely new to an organism, or it could have been experienced within just the last few minutes. In the former case, it is called complete novelty, and in the latter case, it is called short-term novelty. The intermediate case where a stimulus has not been experienced for a period of time (usually days) is called long-term novelty. Based on whether a stimulus possesses qualities that have been perceived before, it is classified into *absolute novelty* or *relative novelty*. If the stimulus does not have any previously perceived quality, then it is absolute novelty; otherwise, it is relative novelty.

Based on these observations, Berlyne introduced three criteria to measure novelty, i.e., novelty is inversely related to (1) how often the stimuli have been experienced before, (2) how recently the stimuli have been experienced, and (3) how similar the stimuli are to previously experienced ones.

Novelty is often accompanied by other properties, each of which may have different influences on exploratory behaviors. Berlyne listed them as supplementary variables to novelty, which include change, surprisingness, and incongruity. A **change** of the stimulus in question may induce some priority in exploratory directions. **Surprisingness** arises when there is a stimulus that induces an expectation and a later stimulus that contradicts the expectation. **Incongruity** is somewhat different from surprisingness (e.g., the statement that the Earth is round is a surprise to people who are accustomed to the concept that the Earth is flat), and indicates an expectation that is not met by the same stimulus (e.g., a person who is used to receiving gifts on his birthday before may find the experience incongruent when nobody sends him birthday gifts this year).

Uncertainty arises when an organism has difficulty selecting a response to a stimulus. Berlyne adopted information theory to quantify uncertainty. He proposed to measure the uncertainty caused by a stimulus with the following steps: (1) draw up a list of stimuli that might occur (as a response to the stimulus in question), (2) partition them into classes, and (3) assign a probability to each class. The probability p_i of each class denotes the competing strength of each possible response, and Shannon's entropy $H = -\sum_{i=1}^n (p_i \times \log_2(p_i))$ denotes the degree of uncertainty.

Conflict occurs when a stimulus arouses two or more incompatible responses in an organism. A response can be incompatible with one another due to different reasons.

Firstly, some responses may be innately antagonist to each other. For example, no organism can move forward and backward at the same time. Secondly, some responses may initially be capable of performing together but become incompatible through learning. For example, we seldom frown when shaking hands. Another reason for incompatible responses may be attributed to the limitation of an organism’s ability to multi-task. For example, it would be considered an outstanding ability if a person can read two books at the same time.

Berlyne proposed four criteria for measuring conflict. Conflict is positively related to (1) the nearness to equality in the strengths of competing responses, (2) the absolute strengths of competing responses, (3) the number of competing responses, and (4) the degree of incompatibility between competing responses.

Complexity roughly refers to the variety or diversity in a stimulus pattern. Three most obvious properties that determine the degree of complexity are: (1) the number of distinguishable elements in a stimulus, (2) the dissimilarity between these elements, and (3) the degree to which several elements are perceived and responded to as a unit.

In summary, collative variables are properties of a stimulus that describe the curiosity stimulating conditions. According to Berlyne, they are all eminently quantitative properties that exist in varying degrees. This endows them potential candidates for measuring the stimulation level of curiosity. Most of the computational models of curiosity have considered at least one of the collative variables for the determination of stimulation level (a review of the same will be provided in Section 2.2.1).

2.1.2.2 Intermediate Arousal Potential

The existence of a stimulus does not necessarily result in curiosity. The arousal of curiosity depends on an appropriate level of stimulation induced by a stimulus. In the 1870s, Wundt [149] introduced the concept of “optimal level of stimulation” and postulated an inverted “U-shape” relationship between the stimulation level and the hedonic value, referred to as the **Wundt curve** (Figure 2.1). It is a general rule stating that many forms of stimulation are pleasant at medium intensities and become unpleasant when their intensities are too high. Based on Wundt’s theory and other experimental results from the literature, Berlyne formed the theory of “intermediate arousal potential”,

Figure 2.1: The Wundt curve (adopted from Berlyne [8]).

where too little stimulation results in boredom, too much stimulation results in anxiety and only intermediate stimulation results in curiosity. The two ends of the spectrum in the Wundt curve reflect two rules in stimulus selection: *Avoidance of Boredom* (**AoB**) and *Avoidance of Anxiety* (**AoA**).

To summarize, from the psychological point of view, the arousal process of curiosity can be abstracted into a **two-step process model**, which offers a uniform angle for examining and analyzing existing computational models of curiosity:

- **Step 1:** evaluation of the stimulation level based on collative variables.
- **Step 2:** evaluation of the interest level based on the principle of intermediate arousal potential.

Next, based on this two-step process model of curiosity arousal, we direct our angle from psychology to computation and review the existing computational models of curiosity and their major applications.

2.2 Computational Curiosity in AI

Inspired by the many advantages of curiosity on human beings, researchers started building computational forms of curiosity to endow agents with desirable functions. In this

Figure 2.2: A general appraisal process of computational curiosity.

section, we review computational curiosity from two aspects: the existing computational models of curiosity and their major applications.

2.2.1 Computational Models of Curiosity

In this section, we review and analyze the existing computational models of curiosity from the perspective of the proposed two-step process model. A general appraisal process of computational curiosity is summarized in Figure 2.2. In this figure, the input stimuli are data samples perceived by the agent. The input stimuli trigger the agent’s computational appraisal process of curiosity. In step 1, the appraisal process first evaluates the level of stimulation elicited by the stimuli, based on collative variables. Some of the existing models adopt a single collative variable to determine the stimulation value (e.g., [108]), while others aggregate multiple collative variables to derive the stimulation value (e.g., [73]).

In step 2, the level of curiosity is evaluated through a mapping from the stimulation value to the curiosity value. Some of the existing models follow the principle of “intermediate arousal potential” by explicitly simulating the Wundt curve, which represents a nonlinear mapping from the stimulation value to the curiosity value (e.g., [108, 79]). These models accommodate both AoB and AoA in their stimulus selection approaches. In comparison, other models simply use the stimulation value as the curiosity value (e.g., [112, 87, 70]). Some of these models also consider the principle of “intermediate arousal potential” when determining the stimulation value (e.g., [112, 87]), which accommodate both AoB and AoA. The rest of these models simply assume a positive correlation between the stimulation value and the curiosity value, which only support AoB but not AoA. In this case, a high stimulation can lead the agent to anxiety (e.g., [70]).

A taxonomy for the existing computational models of curiosity is provided in Table 2.1. The models are classified into five categories according to the collative variables used

for the evaluation of stimulation in Step 1. The five categories include novelty, surprise, uncertainty, change, and complexity. Within each category, the models are further classified into two sub-categories, according to whether the principle of “intermediate arousal potential” is followed in the models. The first sub-category follows this principle and supports both AoB and AoA in stimulus selection, while the second sub-category assumes a positive correlation between the stimulation value and the curiosity value, which only supports AoB. Next, we will discuss each of the models following this taxonomy.

2.2.1.1 Models based on Novelty

To explore the possibility of artificial creativity, Sanders and Gero [108] developed a computational model of curiosity for intelligent design agents, focusing on the appraisal of novelty. The goal of such an agent is to evaluate the interestingness of creative design patterns based on individual experiences, where curiosity is the key for the evaluation of interestingness [108]. In their model, each creative design pattern (generated by a design generator) can be considered as a stimulus and the stimuli experienced in the past are stored in a “conceptual design space”. This “conceptual design space” is modeled by a Self-Organizing Map (SOM), representing the forms that the agent has experienced often enough to learn.

If we view their model from the perspective of the two-step process model, in Step 1, the evaluation of stimulation is governed by the degree of novelty in a stimulus. The evaluation of novelty is achieved by comparing the stimulus (a creative design pattern) with past experiences (the SOM representations), and is defined by the complement of the typicality measure. This implementation is in line with Berlyne’s third criteria for measuring novelty: novelty is inversely proportional to similarity (typicality).

In Step 2, Saunders and Gero adopted the principle of intermediate arousal potential and explicitly modeled the Wundt curve as a non-linear function. This function is defined by the difference of two sigmoid functions as follows [105]:

$$H = R - P$$

$$R = \frac{R_{max}}{1 + e^{-\rho_R(novelty - R_{min})}} \quad (2.1)$$

Table 2.1: A taxonomy of the existing computational models of curiosity in literature

Collative Variables	Avoidance of Boredom & Avoidance of Anxiety	Avoidance of Boredom
Novelty	Similarity & Simulation of the Wundt curve	Similarity Macedo & Cardoso[99]
		Frequency& Similarity Ogino et al.[06]
Surprise	Prediction error in zero sum games	Prediction error Schmidhuber[91a] Uğur et al.[07]
	Information gain	Improbability Macedo & Cardoso[99]
Uncertainty	-	Entropy Macedo & Cardoso[95]
Change	Spread adjustment in Gaussian	-
Complexity	Prediction improvement between successive situations	-
	Prediction improvement between similar situations	
	Compression improvement	
	Discriminability difference between two spaces	

$$P = \frac{P_{max}}{1 + e^{-\rho_P(novelty - P_{min})}}$$

where H is the curiosity value, R , P , R_{max} , P_{max} , ρ_R , ρ_P , R_{min} and P_{min} are the reward, punishment, maximum reward, maximum punishment, slope of the reward sigmoid functions, slope of the punishment sigmoid functions, minimum novelty to be rewarded and minimum novelty to be punished, respectively. This curiosity value is the evaluation of the interestingness of a design pattern or how curious it is about the design pattern.

Later, Merrick and Maher applied this model of curiosity in Non-Player Characters (NPCs) and reconfigurable robots [79, 78, 74, 77, 75] to enhance their performances. For NPCs, the goal of infusing curiosity is to achieve creative behaviors, namely, behavioral diversity [79, 74]. These works are rooted in a motivated reinforcement learning framework, by which NPCs can learn the state-action policy in a virtual environment. Here, each event is regarded as a stimulus and is defined as the change between the current state and the previous one. All previously experienced events are stored in an SOM structure. A habituation layer of neurons is connected to the clustering layer and a habituation function is used to compute novelty. The final curiosity value is obtained by feeding the novelty value into the simulated Wundt curve (Equation 2.1). The curiosity value is used as an intrinsic reward to update the policy that maps states to actions. This curiosity reward can eventually lead to creative behaviors, because each new interesting event results in a perturbation of the existing behavioral patterns and form reward signals for the learning process.

For reconfigurable robots [77], the role of curiosity is to direct the robots' attention to reconfigurations in their structures. This is an important skill for them to learn new behaviors in response to structural changes. For robots, an event can be considered as a stimulus, which is defined as the change between the current sensation and the previous one. Following a similar algorithm in curious NPCs, curiosity rewards are generated for reconfigurable robots. With the curiosity rewards, the robots are self-motivated to explore changes in their structures and develop new behaviors.

In the domain of planetary exploration and map-building of interiors, Macedo and Cardoso [69] proposed a model of curiosity for intelligent agents to simulate human-like exploratory behaviors in unknown environments. Along their research, they gradually

introduced novelty, surprise, and uncertainty into their computational model of curiosity. Here, we first look at their model of novelty.

Macedo and Cardoso’s model relies on graph-based mental representations of objects. Each object can be regarded as a stimulus. In Step 1, the level of novelty regarding a stimulus (object) is measured based on the error correcting code theory of Hamming. Three steps are considered: 1) representing each graph (describing an object) in a common shape matrix 2) extracting the numerical code from the matrix representation of each graph, and 3) computing the Hamming Distance. Here, novelty is defined as the minimum Hamming Distance from the stimulus to the ones that have been experienced before, which determines the stimulation value. In Step 2, the stimulation value is directly used as the curiosity value. The system only supports AoB because the chance of an object to be explored is positively correlated to the level of curiosity, which direct the system away from boredom.

For both the models proposed by Sanders & Gero [108] and Macedo & Cardoso [69], regardless of the implementation, i.e., atypicality in SOM or hamming distance in graph-based mental representations, novelty reflects a comparison between the current stimuli and previous experiences. In other words, similarity. In some later models of curiosity, the factor of time, as another dimension in addition to similarity, is considered for measuring novelty.

Ogino et al. [84] addressed lexical acquisition problems for robots using computational curiosity to associate visual features of observed objects with the labels (for objects) that are uttered by a caregiver. In their work, each object can be interpreted as a stimulus for the robot. Visual features of each object are represented by an SOM. In Step 1, stimulation value is determined by novelty. The novelty of each object is calculated based on two types of saliency: habituation saliency, reflecting the temporal infrequency; and knowledge-driven saliency, reflecting the level of dissimilarity. The habituation saliency is characterized by habituation and inversely related to the frequency of the observation of a visual feature. The knowledge-driven saliency is characterized by the acquired knowledge, where more saliency is given to visual features that are not associated with other labels. The product of the two saliency values represents the overall level of stimulation, which is directed used as curiosity value (in Step 2). The robot chooses to learn about the

object with maximum curiosity value, which drives the system away from boredom and considers the AoB rule in stimulus selection. Their experimental results show that the infusion of curiosity helps accelerate the robot’s learning.

2.2.1.2 Models based on Surprise

Two interpretations for surprise exist in the literature of computational curiosity. The first one interprets surprise as the difference between an expectation and the real outcome. Prediction error matches well with this interpretation and has been utilized in many curiosity models to measure the level of surprise [111, 4, 114, 136]. The second interpretation describes surprise as the degree of not expecting something. Storck et al. [130] modeled this type of surprise using the information gain before and after an observation, while Macedo and Cardoso [70] proposed another measure using improbability. Next, we will discuss each of these models in detail.

Prediction Error-based Models: Schmidhuber [111] introduced artificial curiosity into model building control systems. The goal of such a control system is to learn the input-output mapping in a noisy environment. Curiosity is infused to give the control system an intrinsic desire to improve the model’s knowledge about the world. This is realized by introducing an additional reinforcement unit on top of the controller, which rewards actions that cause high prediction errors. The prediction error inherently measures the degree of mismatch between belief and reality, which is in line with the first interpretation of surprise. In Step 1, the stimulation value is determined by surprise (prediction error) and it is directly used as the curiosity value (in Step 2) to form decisions (on rewarding the system). This mechanism (rewarding the system with high surprise/curiosity) encourages certain past actions to be carried out again in order to repeat situations similar to the mismatched ones. Hence, the system will always direct its attention toward something that is unknown and therefore avoid boredom. This working mechanism only supports AoB in stimulus selection. Barto’s theory of intrinsically motivated reinforcement learning is implemented based on a similar principle [4, 126].

In a later variation, Schmidhuber [114, 115] worked on exploring the space of general algorithms that can automatically create predictable internal abstractions of complex spatio-temporal events. In this model, both AoB and AoA are accommodated. Curiosity

is interpreted as the ability of the system to focus on interesting things by losing interest in the overly predictable (boring) or the overly unpredictable (anxiety inducing) aspects of the world. To achieve this goal, the system is realized by two intrinsically motivated agents playing zero-sum games. The two agents can bet in advance on whether a prediction is true or false. If they bet on different outcomes, the system will check who is right. The winner gets rewarded by receiving the other's bids whereas the loser loses its bids due to surprise (error in prediction). Hence, both agents are motivated to lure the opponent into agreeing on computation sequences that will surprise the other one. However, a surprised module will eventually adapt and in turn, cause the other agent to lose a source of reward. In this way, the system as a whole is motivated to shift the exploration focus and reveal the unknown yet predictable regularities.

Another example of prediction error-based implementation of surprise is given by the work of Uğur et al. [136]. In situations where a robot physically interacts with the environment to explore and learn, assuming the traditional reinforcement learning method is applied, the robot may require a large number of interactions to learn even simple tasks. This could eventually damage the robot. To address this problem, Uğur adopted a Support Vector Machine (SVM) to learn the perception and action mapping in the robot, where curiosity is introduced to select interesting training data for SVM to reduce the total number of training data required. In Step 1, the stimulation level is determined by surprise, which is measured by a sample's distance to the hyperplane (which separates two classes) in the feature space. In Step 2, the stimulation value is directly used as curiosity value for decision making. The system supports AoB because it can be driven away from boredom: only if the distance is smaller than a fixed threshold (curiosity value is high), the sample is considered interesting and sent for learning.

Probability-based Models: An extension of Schmidhuber's curiosity principle to non-deterministic environments was done by Storck et al. [130]. The goal of their system is to learn the model of a non-deterministic Markov environment where each state-action pair $(S(t), a(t))$ may result in different next states $(S(t + 1))$ probabilistically. The goal of introducing curiosity into this learning system is to actively search for interesting training examples that can maximize expected learning improvement. In step 1, the stimulation value is determined by surprise, which reflects "the degree of not expecting something".

They adopt a probabilistic way of measuring surprise, which is the Kullback-Leibler distance between the agent’s belief distribution before and after making an observation. This value of information gain reflects the learnability of the exploration space. Exploration areas where little information gain can be achieved are either too predictable (well-learned) or too unpredictable (inherently unlearnable). Hence, the information gain (surprise) accommodates both AoB and AoA in stimulus selection. In step 2, the surprise value is directly used as the curiosity value, which acts as rewards for the reinforcement learning of state-action pairs. In this way, curiosity drives the system away from the exploration space where little information gain can be achieved, i.e., areas that are either too boring or anxiety-inducing.

Macedo and Cardoso [69, 70, 72, 73], mentioned in Section 2.2.1.1, also modeled human forms of surprise in artificial agents. Surprise serves as an intrinsic drive to direct the agent’s exploration in unknown environment. Their model relies on graph-based mental representations of objects, where each object is considered as a stimulus. The surprise level is defined by the degree of not expecting a stimulus (object), and is implemented as the improbability of existence of the stimulus. The improbability is evaluated based on probability theory: the degree of expecting an event X is given by its probability $P(X)$, and accordingly, the improbability of X , denoted by $1 - P(X)$, defines the degree of not expecting X (surprise). In the case of an object Obj_k comprising several components, the probability of expecting an object is calculated by the mean value of conditional probabilities of its n constituent parts, where each part is individually computed based on Bayes’s formula, given by

$$P(Obj_k) = \frac{\sum_{l=1}^n P(Obj_k^l | Obj_k^1, Obj_k^2, \dots, Obj_k^{l-1}, Obj_k^{l+1}, \dots, Obj_k^n)}{n} \quad (2.2)$$

where $P(Obj_k)$ is the probability of expecting the k^{th} object. However, surprise in this model is not treated as a stimulating factor for curiosity. Hence, the surprise value is not mapped to the curiosity value.

2.2.1.3 Models based on Change

In robot sensory-motor learning, Karaoguz et al. [48] proposed a model of curiosity to direct robots’ attention to changes in the environment. The aim of the model is not to

measure the amount (or level) of changes in a stimulus, but rather to focus on how to react when changes occur. Hence, the system does not provide evaluation for the level of stimulation or the level of curiosity (induced by change), but offers mechanisms to redirect its attention to changes.

For the robot, the focus of attention is determined by a Gaussian distribution that governs the sampling of training examples over the laser motor space. The center of the Gaussian distribution is determined by the mean value of the last N samples added to the mapping, and directs the attention of the robot to areas where new samples have been recently added. The spread of Gaussian distribution is related to the performance (success rate) of the mapping by

$$\sigma(t, s_r) = \sigma_0 + k_b t_b + k_f (t_f - s_r)^2 \quad (2.3)$$

where σ_0 is the baseline, t_b is the number of time steps since the last new sample was added, k_b is the boredom coefficient, t_f is the failure threshold, s_r is the success rate, and k_f is the failure coefficient.

It can be seen from Equation 2.3 that the attention spread is inversely correlated to the success rate. For example, the attention spread is wide when the success rate is low due to a high value of $k_f(t_f - s_r)^2$. When a change happens, t_b will be reset to 0 and the high past success rate of that region will result in a narrow Gaussian distribution of samples. Newly-added links in unlearnable areas (noisy region) will fail on retest, which can drive the success rate for that region below the failure threshold t_f and increase the width of sampling distribution, redirecting the system away from that region. Hence, the system has a drive towards regions where the existing mappings are not making accurate predictions (AoB) and a drive away from the regions where no improvements can be obtained (AoA).

From the perspective of machine learning, one can argue that this model is a special case of the earlier introduced models based on predictors [111, 112], because a predictor always predicts that the next observation is similar to the current observation, which indicates no change. Once a change occurs, the predictor makes an error and updates its knowledge about the world.

2.2.1.4 Models based on Uncertainty

Uncertainty arises when there is no clear response to a stimulus [8]. The entropy in information theory has been proposed to measure the degree of uncertainty in a stimulus. This measure has also been employed very often in computational implementations to realize uncertainty-based curiosity [73]. To improve an agent’s exploration in unknown environments, Macedo and Cardoso [73] introduced a measure of uncertainty, on top of novelty [69, 70, 72], into their model of curiosity. Based on the same system setup as presented in Section 2.2.1.1, Macedo and Cardoso [73] argued that the desire to know or learn an object can be induced by both novelty and uncertainty. Each object (stimulus) can contain known parts (without uncertainty) and uncertain parts. The known parts of an object are used to measure novelty through Hamming Distance (introduced in Section 2.2.1.1), whereas uncertainty is measured by the entropy of all uncertain parts, including analogical and propositional descriptions of the physical structure, and functions of the object. In step 1, the stimulation value is determined by the aggregation of novelty and uncertainty. In step 2, the stimulation value is directly used as curiosity value. The system adopts AoB in stimulus selection and chooses objects with highest curiosity value to explore.

2.2.1.5 Models based on Complexity

In machine learning models of curiosity, complexity has been associated with the predictive power of predictive systems or compressive power of data compression systems [60, 112, 116, 87]. In this subsection, we review these models and their variations.

In Section 2.2.1.2, we introduced Schmidhuber’s early ideas on implementing curiosity by rewarding the system proportionally to surprise, i.e., prediction error [111]. However, this implementation only succeeds in guiding the system to avoid boredom (i.e., well learnt area) but not anxiety (i.e., inherently unpredictable area caused by noisy). Later, Schmidhuber refined the model to accommodate both AoB and AoA. He defined curiosity as a simple principle: to learn a mapping from actions to the expectation of future performance improvement [112, 113]. Instead of pure prediction error, reward is now generated according to the controller’s prediction improvement, i.e., change in prediction error. In step 1, the complexity of a data to be learnt by the system (e.g., familiar data

that is easy to learn or noises that are too difficult to learn) is determined by prediction improvement, which supports both AoB and AoA in stimulus selection. The prediction improvement is obtained by a confidence module, which evaluates the reliability of a prediction and can be realized by probability-based or error-based methods. In step 2, the stimulation value is directly used as curiosity value, which is adopted as intrinsic rewards for learning. In this way, the controller will choose actions (based on the delayed reward) to deliberately select training examples with easily learnable regularities. Hence, the infusion of curiosity can direct the system’s attention away from the exploration space that is either too predictable or too unpredictable.

Schmidhuber [116, 118, 119] formed his formal theory of creativity, by generalizing the simple principle of curiosity from predictors to data compressors. According to this theory, a ‘beautiful’ sensory data is one that is simple yet has not been fully assimilated by the adaptive observer, which is still learning to compress data better. The agent’s goal is to create action sequences that can extend the observation history to yield previously unpredictable but quickly learnable algorithmic regularities. In other words, it is looking for data with high compressibility (reflected by curiosity value).

Schmidhuber’s implementation of prediction improvement by nature is a comparison of prediction error between situations that are successive in time. This principle allows robots to avoid long periods of time in front of a television with white noise (completely unlearnable situations) because the prediction error will remain large and the robot will be bored due to little prediction improvement. However, this principle is not robust enough in the alternation of completely predictable and unpredictable situations, because robots can get stuck here due to large prediction improvements. To cope with such problems, Oudeyer and Kaplan [87, 89] refined Schmidhuber’s simple principle. Instead of comparing the prediction error between situations that are successive in time, they compare the prediction error between situations that are similar. They proposed a model of curiosity that allows a robot to group similar situations into regions where comparison between situations is meaningful. The learning space is divided into regions and each region has an expert to make local predictions. Each expert computes the prediction improvement (curiosity value) locally and rewards its state-action pairs according to the prediction improvement. This works well in practice to handle problems such as robots

get stuck in situations with completely predictable and unpredictable sample data in alternation.

Another variation of using prediction improvement as curiosity drive has been proposed by Pang et al. [90]. This model is rooted in incremental Linear Discriminant Analysis (LDA). Here, curiosity is measured as the discriminability difference (residue) between the LDA transformed space and the original space. The infusion of curiosity can help the system actively search for informative examples to learn and improve the performance using fewer instances.

2.2.1.6 Discussion

An interesting point worth noting is that, from the machine learning perspective, there are certain recurring principles underlying curiosity-inspired algorithms. The first group of recurring principles include generating intrinsic curious rewards based on errors [111, 136] or Shannon's information [123]. This group of principles can redirect learning to focus on the unknown samples. However, they fail to distinguish noise from the novel and learnable regularities. The second group of recurring principles include generating intrinsic curious rewards based on error reduction [112, 87], information gain [130], or compression improvement [116]. This group of principles effectively address the above mentioned problem. They are able to guide learning to focus on easily learnable regularities and at the same time filter out noise. The second group of principles form the basis of Schmidhuber's theory of artificial curiosity, which shows success in speeding up learning and building unsupervised developmental systems [112]. Also, Schmidhuber believes that these principles make machines "creative" and intrinsically motivate machines to create action sequences that make data interesting, which forms the basis of his theory of artificial creativity [116].

2.2.2 Major Applications

In the state-of-the-art literature, computational curiosity has mainly been applied in the following four fields: machines learning, robotics, artificial creativity, and games. In this section, we will discuss these major applications where computational curiosity has been studied.

2.2.2.1 Machine Learning

The close-knit relationship between curiosity and human learning has inspired many researchers to devise computational forms of curiosity for machine learning systems, with the expectation to enhance learning capability and potentially drive them to evolve into autonomous intelligent agents. The study of computational curiosity in machine learning systems has some overlap with other concepts such as “active learning” and “intrinsically motivated learning”.

Active learning, as the name suggests, attempts to machines “active” by allowing the learning system to select actions or make queries that influence what data to be added into its training set [17]. Active learning is especially useful when data are expensive or difficult to obtain. Curiosity can play a critical role in active learning by helping the system to determine which data are interesting. For example, Scott and Markovitch [123] introduced a curiosity drive into supervised learning systems to actively select the most informative samples for learning. Here, the system is continually directed towards regions with highest uncertainty, which is also a general principle followed by many other active learning algorithms [22]. Uğur et al. [136] infused curiosity into an SVM-based learning system to select interesting training data, which significantly reduces the number of training samples required to learn. Similarly, Pang et al. [90] introduced curiosity into an LDA-based learning system.

Intrinsically motivated learning advocates the development of “intrinsic motivations” for learning systems to achieve task-independent learning [4, 126] or autonomous development [87]. These learning approaches are gaining increasing popularity among AI researchers [2]. Intrinsically motivated learning often takes root in a reinforcement learning framework, where intrinsic motivations act as intrinsically generated rewards that are to be maximized. In human psychology, curiosity is known to be one of the most important intrinsic motivations related to learning. Hence, curiosity has often been adopted in intrinsically motivated learning algorithms. For example, Barto et al. [4] used Berlyne’s theory as the psychological foundations to develop his intrinsically motivated learning algorithm. Schmidhuber [112, 114, 119] introduced artificial curiosity as intrinsic rewards for general learning algorithms. Oudeyer and Kaplan[87] proposed an intelligent adaptive curiosity mechanism for intrinsically motivated robots.

2.2.2.2 Robotics

With the attempt to design robots that can autonomously self-develop in a progressive manner, Oudeyer and Kaplan [87] devised for them a mechanism that resembles human curiosity. This mechanism acts as an intrinsic motivation to motivate robots to explore into regions with new knowledge [88], and endows them with the ability to adapt to new environments without prior knowledge or manual adjustment. With a similar goal of designing robots that can self-develop without an explicit teacher, Ngo et al. [82] applied Schmidhuber’s principle of curiosity into a robot arm that enables robots to learn skills through playing. Pape et al. [91] applied the same into a biomimetic robot finger for learning tactile skills.

Traversability affordance refers to the ability of robots to navigate through an environment with obstacles. This ability is highly dependent on the robot’s current location, orientation, and the shape of objects in the environment. In situations where robots physically interact with the environment to explore and learn, assuming traditional reinforcement learning methods are applied, even simple tasks such as avoiding objects may require a large number of trials. This increases the risk of the robot being damaged during the exploration. To address this problem, Uğur et al. [136] simulated curiosity in robots to select informative training samples, which can significantly reduce the number of interactions required with minimal degradations in the learning process.

Another problem in robotics is self-directed reconfiguration. Reconfigurable robots can rearrange their modules to achieve different structures, behaviors, and functions. Instead of looking into how robots can adapt in an unstructured environment, reconfigurable robots focus on adaptation to changes in their own structures and changes of goals when the actuator or effector of the robot changes. Merrick and Huntington [77] introduced curiosity into reconfigurable robots to select informative samples to learn, so that with fewer interactions, robots can still achieve better learning outcomes.

One of the most important sensory-motor problems for robots when interacting with an environment is to learn a mapping from the gaze space (the location of an object) to the reach space (the movement of arms to grasp the object) in response to changes in the environment (camera replacement or changes in the physical environment) without manual recalibration of the hardware. To address this problem, Karaoguz et al. [48]

devised a mechanism of curiosity that drives the exploration into learning spaces where a proper level of complexity is associated with a particular level of capability. With this mechanism, robots can concentrate on highly interesting areas that are neither fully explored nor pure noise. In addition, the mechanism can successfully direct robots' attention to regions where changes have occurred.

Exploration in extreme environments can be a dangerous task for humans. Artificial agents, especially in the form of robots, have been a good substitute for humans to undertake such tasks. Exploration of unknown environments has been an active research field in domains such as planetary exploration, meteorite searches in Antarctic, volcano exploration, and map-building of interiors, etc. [81, 16, 73]. In human beings, exploration is often driven by curiosity, a motivating force for attention focus, determination of interest, and gathering knowledge. Based on these observations, researchers devised artificial forms of curiosity for agent to make proper decisions in unknown environments. For example, Macedo and Cardoso [70, 71, 73] modeled human forms of surprise and curiosity in case-based reasoning frameworks to guide agent's exploration in unknown environments populated with objects. Graziano et al. [29] also discussed the application of computational curiosity to solve autonomous exploration problems.

To summarize, computational curiosity has the potential to contribute to various aspects of robotic systems, such as selective attention, autonomous learning, and self-direction. Numerous studies on computational curiosity are continuously emerging in this field [129, 68, 86].

2.2.2.3 Artificial Creativity

Computational creativity explores the possibility of machines capable of generating creative artifacts that are commonly defined as being previously unknown, useful and surprising [11]. Computational curiosity has been studied in machines to demonstrate creative behaviors.

Based on Csikzentmihalyi's system view of creativity, Saunders [106] postulated two levels of creativity: the individual level and the society level. According to Saunders, there are two questions to be answered at the individual level of creativity: (1) how to evaluate creativity, and (2) how to produce creativity. Saunders and Gero [109] argued

that curiosity can be used to guide problem solving by finding interesting design solutions as well as discovering interesting design problems. Curious design agents were proposed to evaluate the interestingness (creativity) of designs based on novelty. Research of curiosity at the society level looks into the socio-cultural influence of curiosity on creativity and the acceptance of creative works by other individuals. Based on previous works, Saunders [107] studied the society level of creativity by creating a virtual society populated with curious design agents. Simulation results showed that the artificial society exhibited certain similar behaviors as in real human societies.

Schmidhuber [116, 117, 119] explored the relationship between creativity, artists, humor and fun. He argued that art and creativity can be seen as the by-products of curiosity rewards. He further argued that the optimal curious reward framework can be sufficiently formal and precise to allow the implementation on computers and developmental robots. Schmidhuber [116] generalized his simple principle of curiosity to form artificial creativity: "the current compressor of a given subjective observer tries to compress his history of acoustic and other inputs where possible" and "the compression progress becomes the wow-effect or intrinsic reward for the 'creative' action selector, which is thus motivated to create more data that allows for more wow-effect". Later, Schmidhuber [120] proposed a greedy but practical implementation of the basic principle of creativity: POWERPLAY, which can automatically invent or discover problems to train a general problem solver from scratch. In his survey, Schmidhuber [118] drew a comparison between his formal theory with less formal works in aesthetics theory and psychology.

2.2.2.4 Games

With the advances in computer technologies for graphics, processing power, and networking, virtual worlds are emerging as platforms for massive online games. Merrick and Maher [78] highlighted the need for new non-player characters to cope with the increasing complexity and functionality of multi-user virtual worlds. They argued that the behavioral traits of humans and animals generated by curiosity can also advance the performance of artificial agents when dealing with complex or dynamic environments, where only limited information is available and the information changes over time. To cope with the difficulty of predefining task specific rules or environment-specific motivation signals, Merrick et al. [79] introduced a mechanism of curiosity into non-player

characters, which enables them to direct attention to relevant information and be curious about changes in the environment. Simulation results showed that the curious motive led the non-player characters to demonstrate higher variety and complexity in behavior patterns [74].

While most of the works that apply computational intelligence to games focused on the generation of behaviors, strategies, or environments, Togelius and Schmidhuber [134] looked at the very heart of games: the rules that define a game. They proposed automatic game designs based on Koster’s theory of fun and Schmidhuber’s theory of artificial curiosity, where Schmidhuber’s theory of curiosity is a coarse approximation of Koster’s theory of fun [58], i.e., a game is fun if it is learnable but not trivial. Li and Riedl [59] proposed that plotlines in role-playing games can be adapted to elicit curiosity in order to improve the replay value of games, but they did not specify concrete mechanisms for modeling curiosity.

2.3 Summary

In this chapter, we first reviewed the existing theories on curiosity in human psychology. It offers fundamental guidelines for modeling computational curiosity in humanoid agents. Along this review of psychology, we abstracted a two-step process model of curiosity. It serves as a psychological viewpoint to study the computational implementations of curiosity. Subsequently, based on this two-step process model, we reviewed the existing computational models of curiosity and their major applications.

This literature review shows that the existing computational models of curiosity are often built heuristically, mostly as algorithm principles to enhance agents’ learning ability or as novelty detection principles to guide agents finding interesting designs. However, these models cannot be easily transformed to address problems in other domains of application. For example, the machine learning algorithm principles cannot support a virtual learning companion to demonstrate human-like curious behaviors in a human-agent interaction context. Similarly, they cannot drive a recommender agent to detect surprising items that may elicit a user’s curiosity. Hence, it is necessary to build a generic computational model of curiosity for intelligent agents, which can be implemented across

domains to support curious behaviors for agents with different goals. Besides the lack of a generic computational model of curiosity, the major applications where curiosity has been studied are limited to machine learning, robotics, artificial creativity, and games. Other application domains such as pedagogical agents and recommender systems, where curiosity may bring significant impacts, have not been explored.

Based on these observations, in this thesis, we propose a computational framework of curiosity for intelligent agents that is based on the theories in psychology. This computational framework of curiosity is composed of a generic computational model of curiosity, a curiosity-driven learning algorithm, and a curiosity-driven recommendation algorithm. In the second part of this thesis, we will introduce the three components in detail.

Part II

Computational Framework of Curiosity

Chapter 3

Computational Model of Curiosity

In this chapter, we present a computational model of curiosity for intelligent agents. This computational model consists of abstract functions and their interactions between each other, which is inspired by Wooldridge’s work [142]. Representing agent models with abstract functions makes them general enough to allow different implementations in different application contexts. This computational model is the basis for the various implementations that will be presented in the subsequent chapters.

The proposed computational model of curiosity for intelligent agents is built based on the model of a standard agent [142], which is a system situated within an environment and consists of basic functions that sense the environment and act on it to achieve goals. However, to support the mechanism of curiosity, a standard agent is required to go beyond the basic functions and possess other important functions such as memory and learning. Memory stores an agent’s previous experiences that are the basis for evaluating the novelty of new experiences. Learning allows an agent to improve its model of the world and make better predictions of the future outcomes, where the accuracy of predictions are the basis for evaluating the surprisingness of new experiences. According to Berlyne’s theory [8], both novelty and surprisingness are important collative variables that govern the curiosity arousal mechanism in human beings. Based on these functions, curious functions are introduced by transposing Berlyne’s theory [8] and Wundt’s theory [149] from psychology. Curious functions are the meta-level decision-making functions that regulates other functions such as learning and acting of the agent.

Next, we will introduce the functions in the proposed computational model of curiosity for intelligent agents and their interactions between each other. We start with the model

of a standard agent and then present the two important functions that are required to support curiosity: memory and learning. After that, we will introduce the curious functions in detail.

3.1 Standard Agent

This section presents the basic functions that form a standard agent. The description of a standard agent begins with the environment with which the agent interacts. The internal components of a standard agent consist of a set of internal functions that map from one type of internal states to another type of internal states.

3.1.1 Environment

According to the classical definition given by Franklin and Graesser [26], an agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future. Hence, an agent must reside in certain environment and act upon it to achieve goals. An agent's environment can be characterized as a set of possible environmental states, denoted by:

$$E = < S > \tag{3.1}$$

where

$$S = \{s_1, s_2, \dots, s_n\} \tag{3.2}$$

Elements in the angle brackets ' $<>$ ' represent different types of components that constitute the composite factor that appearing before '=' (this format will be followed throughout the thesis). In this case, only the environmental state S is highlighted for the environment E , which means agents only pays attention to environmental states (e.g., reinforcement learning agents). In other cases when agents also pay attention to other components of the environment, such as objects (e.g., exploratory robots), the environment can be denoted by $E = < S, O >$, where S represents a set of environmental states and O represents a set of objects. This can be customized for different types of agents. In this chapter, we focus on the environmental state S for illustration.

3.1.2 Internal Functions

To interact with the environment and achieve goals, a standard agent consists of internal functions that observe, deliberate, and then act upon the environment. In this section, the basic internal functions that form a standard agent will be presented.

An agent's ability to effect changes in the environment is determined by the range of actions that it can perform, denoted by:

$$A = \{a_1, a_2, \dots, a_m\} \quad (3.3)$$

Consequently, an agent can be viewed as an abstract function:

$$\mathbf{AGT} : S \rightarrow A \quad (3.4)$$

which maps environmental states to actions.

The behavior of an environment with respect to an agent's actions can be modeled as a function:

$$\mathbf{E} : S \times A \rightarrow \varphi(S) \quad (3.5)$$

which takes the current environmental state $s \in S$ and an agent action $a \in A$, and maps them to a set of environmental states $\varphi(S)$. If all the sets in the range of $\varphi(S)$ are singletons (i.e., the result of performing any action in any state is a set containing a single member that belongs to S), then the environment is *deterministic*, which means its behavior can be accurately predicted.

The abstract function of an agent $\mathbf{AGT}$ can be decomposed into a set of internal functions, which will be discussed next.

3.1.2.1 Perceiving

The function of perceiving captures an agent's ability to understand information about its environment. The perceiving function can be implemented in hardware agents (e.g., mobile robots) by a video camera or an infra-red sensor, or in software agents (e.g., email assistants) by system commands. An agent's ability to perceive the environment is characterized by a set of percepts:

$$P = \{p_1, p_2, \dots, p_l\} \quad (3.6)$$

Consequently, the perceiving function is described by:

$$\mathbf{P} : S \rightarrow P \quad (3.7)$$

which maps environmental states to the internal percepts of the agent.

3.1.2.2 Decision-Making

The function of decision-making encompasses all of the high-level functions of an agent, including belief revision, goal setting, and plan setting. An agent's ability to take appropriate actions is determined by its ability to make decisions, which can be characterized by a set of decision states:

$$D = \{d_1, d_2, \dots, d_k\} \quad (3.8)$$

Consequently, an agent's decision-making process can be represented by a function:

$$\mathbf{D} : P \rightarrow D \quad (3.9)$$

which maps percepts to decision states.

3.1.2.3 Acting

Acting is the process of translating high-level goals and plans lower-level commands that can be carried out by effectors. The acting process can be represented by a function:

$$\mathbf{A} : D \rightarrow A \quad (3.10)$$

which maps decision states to actions.

3.1.2.4 An abstract agent

With all the internal functions described above, a standard agent can be represented abstractly as a compound function:

$$\begin{aligned} \mathbf{AGT} : S &\xrightarrow{\mathbf{P}} P \xrightarrow{\mathbf{D}} D \xrightarrow{\mathbf{A}} A \\ \mathbf{AGT} : \mathbf{A}(\mathbf{D}(\mathbf{P}(S))) \end{aligned} \quad (3.11)$$

The architecture for a standard agent is illustrated in Figure 3.1. The functions described above are illustrated by circular nodes. Solid arrows represent the flow of state variables between functions.

Figure 3.1: The architecture of a standard agent.

3.2 Memory and Learning

The standard agent described above is a simple reactive agent that acts only on the basis of the current percept, ignoring the rest of the percept history. This type of agent only succeeds when the environment is fully observable. However, in most cases, the environment is only partially observable. For example, an robot's range of perception is limited by the sensors that it possesses. To tackle this problem, an agent usually maintains a model of the environment based on the previous perceptions. This model of the environment describes the part of the world that cannot be seen. However, the initial model may not always succeed when the environment reveals more of itself in front of the agent. This phenomenon requires the agent to be adaptive in order to become more competent than its initial knowledge alone may allow.

These requirements are also the important basis for an agent to become curious. For example, being able to remember past experiences allows the agent to evaluate the novelty of newly encountered stimulus; being able to adapt its initial model of the world allows the agent to make more accurate predictions of the future outcomes, which are important basis for evaluating the surprisingness of new experiences. Here, both novelty and surprisingness are key collative variables that govern the curiosity arousal process [8]. All these requirements point to two important abilities that are desirable by intelligent agents: memory and learning. Next, we will present these two functions and their interactions with other internal functions of an agent.

Figure 3.2: The architecture of an agent with memory.

3.2.1 Memory

Memory stores representations of previous experiences, including a number of percepts, decision states, and actions, denoted by:

$$M = \{P^*, D^*, A^*\} \quad (3.12)$$

where P^* , D^* , and A^* represent the previous percepts, decision states, and actions, respectively.

These previous experiences, if organized structurally (e.g., neural networks, plan structures, concept maps, etc.), can form the agent's model of the environment. This model of the environment helps the belief revision, goal setting, and plan setting in the agent's decision-making process, which can be characterized by the following function:

$$\mathbf{D}_M : P \times M \rightarrow D \quad (3.13)$$

An agent with memory is illustrated in Figure 3.2. In this figure, $*$ refers to the previously experienced percepts, decisions, and actions. It can be seen that memory is located between the processes of perceiving, decision-making and acting.

3.2.2 Learning

Learning enables the agents to initially operate in unknown environments and become more competent than its initial knowledge may allow. The agent's ability to adapt is

determined by its ability to update the model of environment stored in its memory and the high level functions in the decision-making process, which is represented by a set of learning rules (or states):

$$L = \{l_1, l_2, \dots, l_i\} \quad (3.14)$$

The learning rules are triggered depending on the current percepts and the previous decision states stored in memory. For example, prediction errors trigger learning, which are obtained by the differences between the predictions (previous decision states) and the true facts (current percepts). This process can be represented by the following function:

$$\mathbf{L} : P \times M \rightarrow L \quad (3.15)$$

Once the learning rules are triggered (e.g., by newly emerged percepts), they can update the model of environment stored in the agent's memory. For example, a planning agent will incorporate a newly encountered event into its plan structure. This process can be represented by the following function:

$$\mathbf{M} : M \times L \rightarrow M' \quad (3.16)$$

where M' is the updated memory.

When the agent's decision is observed to be not optimal, the agent will trigger the learning rules that update the high level functions in the decision-making process to enhance its decision-making ability. For example, a trend analysis agent implemented by neural networks should update its node selection rules when its prediction differs significantly from the true facts. This process can be represented by the following function:

$$\mathbf{D_L} : P \times M \times L \times D \rightarrow D \quad (3.17)$$

which requires the current percepts and the previous decision states stored in memory to trigger learning rules and update the decision-making process.

A learning agent is illustrated in Figure 3.3. It can be seen that the learning function takes the outputs of the perceiving function and the memory function as its inputs. The outputs of the learning function in turn influence the memory function and decision-making function.

Figure 3.3: The architecture of a learning agent.

3.3 Curious Functions

Curious functions model the agent's curiosity appraisal process. In order to simulate a human-like curiosity appraisal process, our work is based on Berlyne's [8] and Wundt's [149] theories in psychology. According to Berlyne, curiosity is a process of stimulus selection, which is governed by a set of collative variables, such as novelty, surprise, uncertainty, conflict, and complexity. Wundt states that the level of curiosity simulation is closely related to two other emotions: boredom and anxiety. According to Wundt, only optimal stimulation level leads to curiosity whereas too less stimulation results in boredom and too much stimulation results in anxiety. Based on these theories, we derived a two-step process model of curiosity arousal, which has been discussed at the end of Section 2.1.2. Following the two-step process model, we propose two curious functions for intelligent agents: stimulus detection and interest evaluation. Stimulus detection is based on Berlyne's theory and corresponds to the first step in the two-step process model, i.e., evaluation of the stimulation level based on collative variables. Interest evaluation is based on Wundt's theory and corresponds to the second step in the two-step process model, i.e., evaluation of the interest level based on the principle of intermediate arousal potential. Next, we will introduce the two curious functions and then discuss their interactions with other internal functions of an agent in detail.

3.3.1 Stimulus Detection

According to Berlyne’s theory [8], curiosity can be viewed as a process of stimulus selection. The stimuli are characterized by a set of collative variables, which can be represented as follows:

$$V = \{\mathcal{N}, \mathcal{S}, \mathcal{U}, \mathcal{F}, \mathcal{H}, \mathcal{P}\} \quad (3.18)$$

where $\mathcal{N}$, $\mathcal{S}$, $\mathcal{U}$, $\mathcal{F}$, $\mathcal{H}$, and $\mathcal{P}$ represent novelty, surprise, uncertainty, conflict, change, and complexity, respectively.

The detection of these collative variables relies on several internal functions of the agent, including perceiving, memory, and decision-making. For example, novelty detection requires a comparison between the current stimuli (obtained by the perceiving function) and previous experiences (stored in memory). Surprise detection involves a comparison between the agent’s prediction (generated by the decision-making function) and the true facts (obtained by the perceiving function). Uncertainty is triggered when a stimulus is difficult to classify (by the perceiving function). Conflict occurs when a stimulus triggers multiple decision states (by the decision-making function). Change happens when the state of a stimulus changes (observed by the perceiving function). Complexity is judged by the agent’s perceiving function of how much variety or diversity in the stimulus pattern. In summary, the stimulus detection process can be characterized by an abstract function as follows:

$$\mathbf{S} : P \times M \times D \rightarrow V \quad (3.19)$$

Note that it is not necessary to model a complete set of collative variables for an agent to be curious, as each collative variable can stand alone to trigger a person’s curiosity. Hence, a subset of collative variables can always be chosen according to the agent’s functional requirements.

3.3.2 Interest Evaluation

According to Wundt’s theory [149], the arousal of curiosity depends on the appropriate level of stimulation that can be induced by a stimulus. Curiosity arouses only if the stimulation is optimal, whereas too little stimulation results in boredom and too

much stimulation results in anxiety. The set of emotions closely related to curiosity is represented by:

$$Em = \{\mathcal{A}, \mathcal{C}, \mathcal{B}\} \quad (3.20)$$

where $\mathcal{A}$, $\mathcal{C}$, and $\mathcal{B}$ represent anxiety, curiosity, and boredom, respectively.

The process of interest evaluation can be represented by an abstract function as follows:

$$\mathbf{I} : \delta(V) \rightarrow Em \quad (3.21)$$

where $\delta(V)$ returns the stimulation level induced by the collative variables, and the function $\mathbf{I}$ maps the curiosity stimulation level to the interest level (indicated by emotions) based on the Wundt curve (Figure 2.1).

3.3.3 Meta-level Decision-making

The two curious functions, i.e., stimulus detection and interest evaluation, are meta-level decision-making functions that interact with agents' learning function and decision-making function to enhance their performances.

Stimulus detection identifies a set of collative variables, all of which reflect certain knowledge gaps between the agent and the environment, which form the motivation for the agents to learn [65]. Hence, the stimulus detection function outputs collative variables that can guide the learning function of an agent to improve its model of the environment. This process can be characterized by the following functions:

$$\mathbf{L}_V : V \times L \rightarrow L \quad (3.22)$$

The stimulus detection function can also influence the agent's decision-making function based on different collative variables identified. For example, human beings will have different coping strategies when facing with novelty and conflict. Novelty often triggers a human being to observe the stimulus in order to understand it, whereas conflict often triggers a human being to think of some ideas to resolve the conflict. This process can be characterized by the following functions:

$$\mathbf{D}_V : V \times D \rightarrow D \quad (3.23)$$

Figure 3.4: The architecture of a curious agent.

Interest evaluation determines the agent's emotion states based on the level of stimulation. Emotion states often influences a person's learning ability. For example, a human being will refuse to learn when he/she is bored but learns much faster when he/she is curious. Hence, emotions can influence the agent's learning as follows:

$$\mathbf{L}_{\mathbf{Em}} : Em \times L \rightarrow L \quad (3.24)$$

Emotions also influences a person's decision-making ability. For example, during a study in gambling decisions and job selection decisions, unhappy subjects were found to prefer high-risk/high-reward options unlike anxious subjects who preferred low-risk/low-reward options [94]. Hence, emotions can influence the agent's decision-making as follows:

$$\mathbf{D}_{\mathbf{Em}} : Em \times D \rightarrow D \quad (3.25)$$

A curious agent architecture is illustrated in Figure 3.4. The two functions: **S** and **I** highlighted with dashed box are the curious functions. It can be observed that stimulus detection requires the outputs of perceiving, memory, and decision-making functions, which outputs detected collative variables to trigger interest evaluation and influence the agent's learning and decision-making functions. The emotions generated through interest evaluation also influence the agent's learning and decision-making functions.

3.4 Summary

In this chapter, we presented a generic computational model of curiosity for intelligent agents. This computational model consists of abstract functions and their interactions between each other, which allows different implementations in different types of agents. This computational model of curiosity is built based on the model of a standard agent, with two important functions introduced that are required to support curiosity appraisal: memory and learning. Based on these functions, two curious functions, i.e., stimulus detection and interest evaluation, are proposed based on Berlyne's and Wundt's theories from psychology. The curious functions serve as meta-level decision-making functions that enhance an agent's learning ability and decision-making ability.

In the next chapter, we will present a curiosity-driven learning algorithm that realizes the proposed generic computational model of curiosity in a neural learning agent.

Chapter 4

Curiosity-driven Extreme Learning Machine

4.1 Introduction

In the previous chapter, a generic computational model of curiosity was proposed for intelligent agent. In this chapter, we focus on realizing the generic computational model of curiosity in a type of neural learning agent: an extreme learning machine (ELM) based classifier.

An extremely fast learning neural algorithm referred to as extreme learning machine (ELM) has been developed for single-hidden layer feedforward networks (SLFNs) by Huang et al. [37, 38]. The essence of ELM is that the hidden layer of SLFNs need not be tuned [36]. ELM randomly assigns hidden neuron parameters and finds the output weights analytically. It has been shown to generate good generalization performance at extremely high learning speed [37, 38, 39, 62] and has been successfully applied to many real world applications [39, 62, 150, 152].

Although ELM has shown advanced generalization performance with extremely high learning speed, several major issues still remain in ELM:

1. Manually set the number of hidden neurons: The number of hidden neurons needs to be set a priori to training [23]. The number of hidden neurons is usually chosen by trial-and-error.
2. Fixed structure: The network structure is fixed once the number of hidden neurons

is set [101]. It cannot evolve, i.e., add or delete hidden neurons, based on the training data.

3. Randomization effect: The random assignment of hidden neuron parameters induces high randomization effect in the generated results.

To address issue 1), several algorithms have been proposed, such as incremental ELM (I-ELM) [34], enhanced incremental ELM (EI-ELM) [35], pruning ELM (P-ELM) [101], optimally-pruned ELM(OP-ELM) [80], and error minimized ELM (EM-ELM) [23]. However, all these algorithms can either add neurons (I-ELM, EI-ELM, EM-ELM) or delete neurons (P-ELM, OP-ELM) without being able to adjust network structure based on the incoming data. In other words, they lack the evolving capability. Recently, a meta-cognitive ELM (McELM) has been proposed [110], which addresses issue 1) and partially issue 2). McELM can decide network structure based on the training data, but it can only add neurons without pruning capability. To our knowledge, few works have been done towards issue 3).

To address all the three issues mentioned above, we propose a curious extreme learning machine (C-ELM) algorithm for classification problems. It is a psychologically inspired algorithm based on the theory of curiosity [145]. In psychology, curiosity is commonly known as the important intrinsic motivation that drives human exploration and learning [65]. The psychological concept of curiosity has been applied in many computational systems to enhance their learning capability (e.g., intrinsically motivated reinforcement learning) [4, 118] and believability (e.g., curious companions) [148, 146, 147]. This is the first attempt to introduce curiosity in an ELM framework.

C-ELM is inspired by the psychological theory of curiosity proposed by Berlyne [8]. Berlyne interpreted curiosity as a process of stimulus selection, i.e., when several conspicuous stimuli are introduced at once, to which stimulus will human respond. He identified several key collative variables, e.g., *novelty*, *uncertainty*, *conflict*, and *surprise*, that govern the stimulus selection process. Based on this theory, C-ELM classifier treats each training data as a stimulus and decides its learning strategy based on the appraisal of collative variables. There are three learning strategies for C-ELM classifier: *neuron addition*, *neuron deletion*, and *parameter update*.

When a new neuron is added, conventional incremental ELM algorithms will randomly assign the center and impact factor of the RBF kernel (other kernels such as linear kernel can also apply) in the new neuron. However, random center selection may require more number of hidden neurons to approximate the decision function accurately [132]. Hence, C-ELM uses data-driven center selection which adopts the current training data that triggers the *neuron addition* strategy as the center of the new neuron. It removes partially the random effect of the traditional ELM algorithms. Data-driven center selection also allows the class label of the new neuron to be apparent, which enables further analysis of the hidden neurons. During *neuron deletion*, the most conflicting neuron for the current training data is removed from the network. In literature, various neuron deletion schemes for ELM have been proposed such as pruning based on relevance [101] or based on leave-one-out cross-validation [80]. These techniques, although effective, might render the system slow. Hence, we propose the *neuron deletion* strategy based on conflict resolution, which helps the system attain fast and efficient convergence. The *parameter update* is conducted using recursive least squares method.

In the rest of this chapter, we will present the detailed definition of the C-ELM classifier and evaluate its performance against other popular classifiers on benchmark data sets.

4.2 Curious Extreme Learning Machine Classifier (C-ELM)

In this section, we provide a detailed description of the C-ELM classifier. The goal of C-ELM classifier is defined as follows:

Given: a stream of training data $\{(\mathbf{x}^1, c^1), \dots, (\mathbf{x}^t, c^t), \dots\}$, where $\mathbf{x}^t = [x_1^t, \dots, x_M^t]^T \in \mathbb{R}^M$ is a M -dimensional input vector of the t th input data, $c^t \in [1, 2, \dots, N]$ is its class label, and N represents the total number of distinct classes. The coded class label $\mathbf{y}^t = [y_1^t, \dots, y_j^t, \dots, y_N^t] \in \mathbb{R}^N$ is obtained by converting the class label (c^t) as follows:

$$y_j^t = \begin{cases} 1 & \text{If } j = c^t \\ -1 & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, N. \quad (4.1)$$

Find: a decision function $\mathbb{F}$ that maps the input features ($\mathbf{x}^t$) to the coded class labels ($\mathbf{y}^t$), i.e., $\mathbb{F} : \mathbb{R}^M \rightarrow \mathbb{R}^N$, as close as possible.

To solve this problem, C-ELM employs two major components: an internal cognitive component which is a unified single layer feed-forward neural network (SLFN) and a curiosity appraisal component that consists of curious functions that regulate the extreme learning process.

4.2.1 The Internal Cognitive Component: SLFN

The internal cognitive component of the C-ELM is an SLFN with M input neurons, K hidden neurons and N output neurons. For RBF hidden neuron with activation function $g(x) : \mathfrak{R} \rightarrow \mathfrak{R}$ (e.g., Gaussian), the output of the k th hidden neuron with respect to the input $\mathbf{x}^t$ is given by:

$$G(\mathbf{x}^t, \mathbf{a}_k, b_k) = g(b_k ||\mathbf{x}^t - \mathbf{a}_k||), b_k \in \mathfrak{R}^+, \quad (4.2)$$

where $\mathbf{a}_k$ and b_k are the center and impact factor of the k th RBF neuron. $\mathfrak{R}^+$ indicates the set of all positive real values.

The predicted output for the input $\mathbf{x}^t$ is given by:

$$\hat{\mathbf{y}}^t = [\hat{y}_1^t, \dots, \hat{y}_i^t, \dots, \hat{y}_N^t]. \quad (4.3)$$

Here, the output of the i th neuron in the output layer is given by:

$$\hat{y}_i^t = \sum_{k=1}^K G(\mathbf{x}^t, \mathbf{a}_k, b_k) w_{ki}, \quad (4.4)$$

where w_{ki} is the output weight connecting the k th hidden neuron to the i th output neuron. The output for a chunk of t input data can be written by:

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{W}, \quad (4.5)$$

where $\mathbf{H}$ is the hidden layer output matrix and $\mathbf{W}$ is the weight matrix connecting the hidden neurons to the output neurons as shown below:

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{x}^1, \mathbf{a}_1, b_1) & \cdots & G(\mathbf{x}^1, \mathbf{a}_K, b_K) \\ \vdots & \ddots & \vdots \\ G(\mathbf{x}^t, \mathbf{a}_1, b_1) & \cdots & G(\mathbf{x}^t, \mathbf{a}_K, b_K) \end{bmatrix}_{t \times K}, \quad (4.6)$$

and

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K,1} & w_{K,2} & \cdots & w_{K,n} \end{bmatrix}_{K \times N}. \quad (4.7)$$

With the cognitive component of C-ELM described above, next, we will introduce the curious functions that regulate the extreme learning process.

4.2.2 Curious Functions

C-ELM employs an intrinsically motivated learning paradigm transposed from the psychological theory of curiosity proposed by Berlyne [8]. Learning is regulated based on the curiosity appraisal of input data.

4.2.2.1 Stimulus Selection

For each input data, the stimulus selection is governed by four collative variables, i.e., *novelty*, *uncertainty*, *surprise*, and *conflict*. In this section, we will introduce the definitions of the four collative variables in C-ELM.

Novelty: Novelty reflects how much the input data differs from the network's current knowledge. In kernel methods, spherical potential is often used to determine the novelty of data [131]. The spherical potential of an input data $\mathbf{x}^t$ is defined by (a detailed derivation can be found in [131]):

$$\psi(\mathbf{x}^t) = \frac{1}{K} \sum_{k=1}^K G(\mathbf{x}^t, \mathbf{a}_k, b_k). \quad (4.8)$$

A higher potential indicates that the input data is more similar to the existing knowledge, while a smaller potential indicates that the input data is more novel. Hence, the novelty $\mathcal{N}$ of an input data $\mathbf{x}^t$ is determined by:

$$\mathcal{N}(\mathbf{x}^t) = 1 - \frac{1}{K} \sum_{k=1}^K G(\mathbf{x}^t, \mathbf{a}_k, b_k). \quad (4.9)$$

Uncertainty: Uncertainty reflects how not confident the network is in its predictions. The confidence of a network is often measured by the posterior probability of the prediction. It has been proven theoretically that hinge-loss function can accurately estimate

the posterior probability for a classification problem [154]. Hence, we use the truncated hinge-loss error ($\mathbf{e}^t = [e_1^t, \dots, e_j^t, \dots, e_N^t]^T \in \Re^N$) [132] to measure the prediction error, where each element is defined by:

$$e_j^t = \begin{cases} 0 & \text{if } \hat{y}_j^t y_j^t > 1 \\ \min(\max(\hat{y}_j^t - y_j^t, -1), 1) & \text{otherwise.} \end{cases} \quad (4.10)$$

With the truncated hinge-loss error, the posterior probability of input $\mathbf{x}^t$ belonging to class c is given by:

$$p(c|\mathbf{x}^t) = \frac{e_j^t + 1}{2}, c = 1, 2, \dots, n. \quad (4.11)$$

Since uncertainty measures how not confident a network is in its predictions, we define uncertainty $\mathcal{U}$ of the prediction to an input data $\mathbf{x}^t$ by:

$$\mathcal{U}(\mathbf{x}^t) = 1 - p(\hat{c}|\mathbf{x}^t), \quad (4.12)$$

where $\hat{c}$ is the predicted class for $\mathbf{x}^t$.

Conflict: In psychology, conflict occurs when a stimulus arouses two or more incompatible responses in an organism [145]. The degree of conflict depends on the competing strengths of those incompatible responses. For a classifier, conflict can be reflected by the competing strengths of the most fired two output neurons.

Given an input $\mathbf{x}^t$, let $\hat{y}_{j_1}^t$ and $\hat{y}_{j_2}^t$ be the outputs of the most fired output neuron and the second most fired two output neurons, respectively. The more closer $\hat{y}_{j_1}^t$ is to $\hat{y}_{j_2}^t$, the higher competing strength is between the two output neurons, which indicates a higher conflict between the network's decisions. Hence, the conflict $\mathcal{F}$ induced by an input $\mathbf{x}^t$ is defined by:

$$\mathcal{F}(\mathbf{x}^t) = \begin{cases} 1 - \frac{|\hat{y}_{j_1}^t - \hat{y}_{j_2}^t|}{|\hat{y}_{j_1}^t + \hat{y}_{j_2}^t|} & \text{if } \hat{y}_{j_1}^t \hat{y}_{j_2}^t > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (4.13)$$

Surprise: In psychology, surprise indicates a violation of expectation [145]. For a classifier, surprise occurs when the predicted output differs from the true class label. The degree of surprise is determined by prediction errors for both the true class and the predicted class. As we adopt hinge-loss error in this work to measure prediction error, the surprise $\mathcal{S}$ induced by an input $\mathbf{x}^t$ is defined by:

$$\mathcal{S}(\mathbf{x}^t) = \begin{cases} |e_c^t \cdot e_{\hat{c}}^t| & \text{if } \hat{c} \neq c \\ 0 & \text{otherwise,} \end{cases} \quad (4.14)$$

where $\hat{c}$ and c represent the predicted class and the true class, respectively; and e is the hinge-loss error calculated according to Eq (4.10).

It can be analyzed that all the four collative variables are within the range of $[0, 1]$. The collative variables determine the level of curiosity arousal and the learning strategy selection. With the collative variables defined as above, we will introduce the learning strategies in the following section.

4.2.2.2 Learning Strategies

C-ELM has three learning strategies: *neuron addition*, *neuron deletion*, and *parameter update*. C-ELM begins with zero hidden neurons, add or delete hidden neurons and update parameters of the existing neurons to achieve an optimal network structure with optimal parameters. The decision on whether to update network structure or update parameters is made based on the appraisal of collative variables. Intuitively, higher values of collative variables induce a higher level of curiosity towards the input data, which require more efforts in learning, i.e., updating network structure, to incorporate the new knowledge; otherwise, simply update parameters of the existing neurons to reinforce the ‘familiar’ knowledge. Next, we will introduce the three learning strategies of C-ELM in detail.

Neuron Addition Strategy: Intuitively, for an input data, if novelty is high and uncertainty is high and surprise is high, it indicates that a misclassification (i.e., surprise high) with high uncertainty in its prediction (i.e., uncertainty high) is caused by the newness of the knowledge (i.e., novelty high). In this case, the network should add new neurons to capture this new knowledge. Hence, given an input $\mathbf{x}^t$, the neuron addition condition is:

$$\mathcal{N}(\mathbf{x}^t) > \theta_{\mathcal{N}^{add}} \text{ AND } \mathcal{U}(\mathbf{x}^t) > \theta_{\mathcal{U}} \text{ AND } \mathcal{S}(\mathbf{x}^t) > \theta_{\mathcal{S}}, \quad (4.15)$$

where $\theta_{\mathcal{N}^{add}}$, $\theta_{\mathcal{U}}$, and $\theta_{\mathcal{S}}$ are neuron addition thresholds for novelty, uncertainty, and surprise, respectively. If these parameters are chosen close to 1, then very few input data can trigger the neuron addition strategy and the network cannot approximate the decision function accurately. If these parameters are chosen close to 0, then many input data can trigger the neuron addition strategy, leading to poor generalization ability. In

general, $\theta_{\mathcal{N}^{add}}$ is chosen in the range of $[0.1, 0.5]$, $\theta_{\mathcal{U}}$ is chosen in the range of $[0.1, 0.3]$, and $\theta_{\mathcal{S}}$ is chosen in the range of $[0.2, 0.9]$.

A typical ELM randomly chooses hidden neuron parameters and finds the output weights analytically. However, random exploration of the feature space may need more hidden neurons to accurately approximate the decision function. In C-ELM, we propose data-driven center selection for the hidden neurons without compromising the extreme learning capability. When a new neuron is added, instead of randomly assigning the center $\mathbf{a}_k$, we assign the input features of the current input data as the center, i.e., $\mathbf{a}_k = \mathbf{x}_t$. Since the center selection is data-driven, we can label the class of the new neuron using the target value of the input data, i.e., $l_k = c^t$. Data-driven center selection allows fast hidden neuron clustering using their class labels and provides class specific information when deleting neurons. The values of the impact factors in the hidden neurons are randomly assigned. Hence, with the new hidden neuron, the dimension of the hidden layer output matrix $\mathbf{H}$ increases from $(t-1) \times (k-1)$ to $t \times k$. The target values of the t input data is represented by:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^t \end{bmatrix}_{t \times N}. \quad (4.16)$$

The output weight for $\mathbf{W}$ can be analytically find by:

$$\mathbf{W} = \mathbf{H}^\dagger \mathbf{Y}, \quad (4.17)$$

where $\mathbf{H}^\dagger$ is the Moore-Penrose generalized inverse of the hidden layer output matrix $\mathbf{H}$.

Neuron Deletion Strategy: Intuitively, for an input data, if surprise is high and conflict is high and novelty is low, it indicates that a misclassification (i.e., surprise high) occurs for a familiar stimulus (i.e., novelty low) due to high competing strengths between two decisions (i.e., conflict). In this case, the network should adjust its decision-making by strengthening the correct decision and weakening the wrong decision, i.e., deleting the most contributing neuron in the wrong class. Hence, given an input $\mathbf{x}^t$, the neuron deletion condition is:

$$\mathcal{S}(\mathbf{x}^t) > \theta_{\mathcal{S}} \text{ AND } \mathcal{F}(\mathbf{x}^t) > \theta_{\mathcal{F}} \text{ AND } \mathcal{N}(\mathbf{x}^t) < \theta_{\mathcal{N}^{del}}, \quad (4.18)$$

where θ_S , θ_F , and $\theta_{N^{del}}$ are neuron deletion thresholds for surprise, conflict, and novelty, respectively. When θ_S and θ_F are chosen close to 1 and $\theta_{N^{del}}$ is chosen close to 0, then very few input data can trigger neuron deletion, leading to poor generalization ability. When θ_S and θ_F are chosen close to 0 and $\theta_{N^{del}}$ is chosen close to 1, then many input data can trigger neuron deletion and the network cannot approximate the decision function accurately. In general, θ_S is chosen in the range of $[0.2, 0.9]$, θ_F is chosen in the range of $[0.1, 0.3]$, and $\theta_{N^{del}}$ is chosen in the range of $[0.1, 0.8]$.

When neuron deletion is triggered, C-ELM will remove the most fired hidden neuron k belonging to the predicted class:

$$\{k | \max(G(\mathbf{x}^t, a_k, b_k)) \text{ AND } l_k = \hat{c}\}. \quad (4.19)$$

After the k th neuron is removed, the network will re-calculate the output weight $\mathbf{W}$ with the t input data.

Parameter Update Strategy: When both the neuron addition strategy and the neuron deletion strategy are not triggered, it indicates that the new input data is a ‘familiar’ one. Hence, the network will update the output weight using recursive least squares to reinforce the familiar knowledge.

For the new input data $\mathbf{x}^t$, let the partial hidden layer output be represented by $\mathbf{h}^t = [G(\mathbf{x}^t, \mathbf{a}_1, b_1), \dots, G(\mathbf{x}^t, \mathbf{a}_K, b_K)]$. The output weights are updated according to [61] by:

$$\mathbf{W}^t = \mathbf{W}^{t-1} + \mathbf{P}^t \mathbf{h}^t ((\mathbf{y}^t)^T - (\mathbf{h}^t)^T \mathbf{W}^{t-1}), \quad (4.20)$$

where

$$\mathbf{P}^t = \mathbf{P}^{t-1} - \frac{\mathbf{P}^{t-1} \mathbf{h}^t (\mathbf{h}^t)^T \mathbf{P}^{t-1}}{1 + (\mathbf{h}^t)^T \mathbf{P}^{t-1} \mathbf{h}^t}. \quad (4.21)$$

A pseudo-code for the C-ELM learning algorithm is given in Algorithm 1. This algorithm involves four steps: in step 1, a new sample is presented for study; step 2 performs curiosity appraisal towards this new sample to compute values for the four collative variables, i.e., novelty, uncertainty, conflict and surprise; step 3 selects a learning strategy from neuron addition, neuron deletion, and parameter update, based on the curiosity appraisal results of the collative variables; and step 4 moves to the next sample to be studied.

Algorithm 1: PSEUDO-CODE FOR C-ELM CLASSIFIER

```
1 Step 1. Present a sample  $(\mathbf{x}^t, c^t)$ .
2 Step 2. Perform curiosity appraisal towards the sample:
3 compute novelty  $\mathcal{N}(\mathbf{x}^t)$  using Eq (4.9), compute uncertainty  $\mathcal{U}(\mathbf{x}^t)$  using Eq
  (4.12), compute conflict  $\mathcal{F}(\mathbf{x}^t)$  using Eq (4.13), and compute surprise  $\mathcal{S}(\mathbf{x}^t)$  using
  Eq (4.14).
4 Step 3. Select learning strategy based on the collative variables:
5 if  $\mathcal{N}(\mathbf{x}^t) > \theta_{\mathcal{N}^{add}}$  AND  $\mathcal{U}(\mathbf{x}^t) > \theta_{\mathcal{U}}$  AND  $\mathcal{S}(\mathbf{x}^t) > \theta_{\mathcal{S}}$  then
6   add hidden neuron with  $\mathbf{x}^t$  as the center and randomize the impact factor of
   the hidden neuron. Update the hidden neuron output matrix  $\mathbf{H}$  and compute
   the output weight  $\mathbf{W}$  according to Eq (4.17).
7 if  $\mathcal{S}(\mathbf{x}^t) > \theta_{\mathcal{S}}$  AND  $\mathcal{F}(\mathbf{x}^t) > \theta_{\mathcal{F}}$  AND  $\mathcal{N}(\mathbf{x}^t) < \theta_{\mathcal{N}^{del}}$  then
8   delete the most fired hidden neuron that belonging to the predicted class
   following Eq (4.19).
9 else
10  update the network parameter using Eq (4.20).
11 end
12 Step 4. Increment  $t = t + 1$ , go to Step 1.
```

4.3 Performance Evaluation of C-ELM

The performance of C-ELM is evaluated on the benchmark problems described in Table 4.1 from the UCI machine learning repository, which contains four multi-category classification problems (vehicle classification, iris, wine, and glass identification) and four binary classification problems (liver disorder, PIMA, breast cancer, and ionosphere). The performance of C-ELM is evaluated in comparison with other popular classifiers such as SVM, ELM and McELM. The results of SVM, ELM and McELM are reproduced from [110]. For simulating the results of C-ELM, MATLAB 2014b with 3.2 ghz and 16gb ram was used. The parameters were optimized using grid search.

4.3.1 Performance Measures

The performance of C-ELM is measured against other popular classifiers using two types of performance measures: average classification accuracy and overall classification accuracy. When the number of samples in each class is highly unbalanced, the average classification accuracy tends to yield more useful information.

Table 4.1: The specification of benchmark data sets on classification problems

Data set	# Features	# Classes	# Training Data	# Testing Data
Vehicle classification	18	4	424	422
Iris	4	4	45	105
Wine	13	3	60	118
Glass identification	9	6	109	105
Liver disorder	6	2	200	145
PIMA	8	2	400	368
Breast cancer	9	2	300	383
Ionosphere	34	2	100	251

Average classification accuracy: The average classification accuracy η_a is defined by:

$$\eta_a = \frac{1}{C} \sum_i^C \frac{q_i}{N_i} \times 100, \quad (4.22)$$

where q_i is the number of data in class i that have been correctly classified, and N_i is the total number of data in class i . It reflects the average ratio of correctly classified data in each class.

Overall classification accuracy: The overall classification accuracy η_o is defined by:

$$\eta_o = \frac{\sum_i^C q_i}{N_T} \times 100 \quad (4.23)$$

where N_T is the total number of data in the testing data set. It reflects the overall ratio of correctly classified data in the whole testing data set.

4.3.2 Performance Study on Multi-category Classification Problems

The performance of the C-ELM on multi-category benchmark classification problems is shown in Table 4.2. It can be observed from Table 4.2 that the generalization performance of C-ELM is better than other classifiers used for comparison on all the multi-category classification problems. Also, the number of hidden neurons added during the evolving process is comparable with other algorithms. For example, C-ELM selected 140 hidden

Table 4.2: Performance comparison between C-ELM and other popular classifiers on the multi-category classification problems

Data set	classifier	# hidden neurons	Training time(s)	Testing	
				η_o	η_a
Vehicle classification	SVM	340 ^a	550	70.62	68.51
	ELM	150	0.4	77.01	77.59
	McELM	120	40	81.04	81.3
	C-ELM ^b	140	15.2	81.99	82.42
Iris	SVM	13 ^a	0.02	96.19	96.19
	ELM	10	0.01	96.19	96.19
	McELM	6	0.03	98.1	98.1
	C-ELM ^c	6	0.0001	99.05	99.05
Wine	SVM	13 ^a	0.1	97.46	98.04
	ELM	10	0.25	97.46	98.04
	McELM	9	0.02	98.31	98.69
	C-ELM ^d	8	0.015	99.15	99.35
Glass identification	SVM	183 ^a	320	70.47	75.61
	ELM	80	0.05	81.31	87.43
	McELM	72	3.2	82.86	87.4
	C-ELM ^e	54	2.87	80.95	90.81

(a) # support vectors. Number of neuron deleted: (b) 6 (c) 0 (d) 0 (e) 8.

neurons with 6 times neuron deletion for Vehicle classification problem, 6 hidden neurons for Iris problem, 8 hidden neurons for Wine problem, and 52 hidden neurons with 8 times neuron deletion for Glass identification problem. Another advantage of C-ELM in comparison with other self-regulated learning algorithms such as McELM is that it takes a substantially small amount of time for training. For example, McELM takes 40 seconds to train the Vehicle classification problem whereas C-ELM only takes 15.2 seconds. Hence, it shows that C-ELM achieves better performance than other classifiers on multi-category classification problems due to the intrinsically motivated learning mechanism of curiosity.

4.3.3 Performance Study on Binary Classification Problems

The performance of C-ELM on three binary classification problems is shown in Table 4.3. Table 4.3 shows that C-ELM achieves better generalization performance than other classifiers used for comparison on all the binary classification problems. Also, the total number of hidden neurons added during the evolving process is comparable with other

Table 4.3: Performance comparison between C-ELM and other popular classifiers on the binary classification problems

Data set	classifier	# hidden neurons	Training time(s)	Testing	
				η_o	η_a
Liver disorder	SVM	141 ^a	0.1	71.03	70.21
	ELM	100	0.17	72.41	71.41
	McELM	50	0.95	74.48	73.83
	C-ELM ^b	31	0.73	76.55	76.5
PIMA	SVM	221 ^a	0.21	77.45	76.33
	ELM	400	0.29	76.63	75.25
	McELM	25	0.47	80.43	78.49
	C-ELM ^c	33	1.42	81.25	80.31
Breast cancer	SVM	24 ^a	0.11	96.6	97.06
	ELM	66	0.14	96.36	96.5
	McELM	10	0.05	97.39	97.84
	C-ELM ^d	9	0.09	97.65	98.04
Ionosphere	SVM	43 ^a	0.02	91.24	88.51
	ELM	32	0.04	89.64	87.52
	McELM	18	0.1	94.82	93.76
	C-ELM ^e	17	0.26	95.22	95.54

(a) # support vectors. Number of neuron deleted: (b) 0 (c) 0 (d) 0 (e) 0.

algorithms. For example, C-ELM requires 31 hidden neurons for Liver disorder problem, 33 hidden neurons for PIMA problem, 9 hidden neurons for Breast cancer problem, and 17 hidden neurons for the Ionosphere problem. For binary classification problems, the training time of C-ELM is comparable with other self-regulated learning algorithms such as McELM. For example, it requires 0.73 seconds for C-ELM and 0.95 seconds for McELM to train the Liver disorder problem. Hence, it shows that C-ELM achieves better generalization ability than other classifiers on binary classification problems without compromising the extreme learning ability of ELM, due to the intrinsically motivated learning mechanism of curiosity.

4.4 Summary

In this chapter, we have presented a curious extreme learning machine (C-ELM) classifier, which is a neural learning agent, based on the generic computational model of curiosity. C-ELM treats each input data as a stimulus for curiosity and performs curiosity

appraisal towards each input data based on four collative variables: *novelty*, *uncertainty*, *conflict*, and *surprise*. Three learning strategies can be chosen from based on the curiosity appraisal results, including *neuron addition*, *neuron deletion*, and *parameter update*. C-ELM enhances traditional ELM algorithms with the evolving capability, which determines optimal network structure dynamically based on the training data. Also, C-ELM reduces partially the random effect of the traditional ELM algorithm by selecting RBF centers based on data instead of random assignment. Moreover, C-ELM employs a novel neuron deletion strategy which is based on conflict resolution. Empirical study of C-ELM shows that the proposed approach leads to compact network structures and generates better generalization performance with fast response, comparing with traditional ELM and other popular classifiers.

In the next chapter, we will present a curious recommendation algorithm which realizes the generic computational model of curiosity in a recommender agent. In the domain of recommender systems, curiosity may allow an agent to recommend interesting items that can elicit the target users' curiosity and motivate them to purchase those items, which may benefit both the business providers and the users at the same time.

Chapter 5

Curiosity-driven Recommendation

5.1 Introduction

In the previous chapter, we presented a curiosity-driven learning algorithm that realizes the generic computational model of curiosity (proposed in Chapter 3) in a learning agent. In this chapter, we focus on the domain of recommender systems and present a curiosity-driven recommendation algorithm that realizes the generic computational model of curiosity in another type of intelligent agents: the social recommender agents.

In the current age of information overload, recommender system has become an indispensable technique for filtering and recommending online information. Due to their effectiveness in helping users filtering through the enormous number of items and in helping enterprisers increasing their sales, recommender systems have been successfully adopted by a number of industrial companies, including but not limited to Amazon, Netflix, Yahoo!News, Apple iTunes, etc.

As highlighted in [99], the ultimate goal of a recommender system is to suggest particularly interesting items, in addition to indicating those that should be filtered out. Traditional recommender systems are built based on a general consensus that user preferences reflect their underlying interests. Hence, various collaborative filtering techniques [104] have been proposed to discover items that best match users' preferences. However, determining interestingness based on user preferences alone is far from sufficient. According to the psychology of interest [125], the appraisal of interestingness in human beings is closely related to curiosity. Instead of focusing on a person's preferences, curiosity is more related to the novelty and surprisingness in the environment. In this work, we take a new

angle to look at the interestingness of recommendation and introduce a novel dimension of social information into the traditional recommender systems: social curiosity.

In real life, it is a commonly observed phenomenon that a person gets curious about the surprising behaviors of his/her friends. For example, Alice knows that her friend Bob always hates horror movies, and the incidence that Bob gives a high rating to a horror movie would interest Alice. In order to find out why Bob gave this surprising rating, Alice may be driven by curiosity to watch this horror movie. This phenomenon is generally known as *social curiosity* in human psychology. It is the desire to acquire new information about how other people behave, think, and feel [98, 145]. Based on this theory, an item rated by a user’s friends can become a recommendation candidate according to how surprising those friends’ ratings are for this item.

Motivated as above, we propose a social curiosity inspired recommender system. On top of user preferences, the interestingness of an item is evaluated based on user curiosity as well. The major contributions of this work are summarized as follows. Firstly, we identify a novel dimension of social information for recommender systems: the social curiosity. Secondly, we build a general and parameter-free model for measuring user curiosity in the social contexts. This model takes into consideration the different responses given by a user to different friends’ surprising ratings. We also propose and compare three strategies for evaluating user curiosity when multiple friends give surprising ratings to the same item. Thirdly, the model is comprehensively studied with two large scale real world datasets: Douban and Flixster. The experiment results show that social curiosity significantly improves recommendation diversity and coverage, while maintaining a sufficient level of accuracy. To the best of our knowledge, this is the first work to explore social information for enhancing recommendation diversity and coverage in recommender systems. In the rest of this chapter, we will implement the proposed computational model of curiosity in recommender agents and conduct experimental studies to analyze the effects brought by curiosity.

5.2 Curious Recommender Agent

In this section, we present how the generic computational model of curiosity (proposed in Chapter 3) is implemented in recommender agents. The goal of a curious recommender

Figure 5.1: The environment of the curious recommender agent.

agent is to recommend such items that both match the target user's preferences and are surprising, and thus stimulate the target user's curiosity. Next, we will present the implementation of the curious recommender agent in detail. We start with the representation of the environment, and then introduce the basic internal function that allow the agent to predict the target user's ratings for unexperienced items. After that, we introduce the curious functions that enable the agent to detect surprising items that may interest the target user.

5.2.1 Environment

The environment where the curious recommender agent resides is illustrated in Figure 5.1. It can be seen that the curious recommender agent observes the user social network and users' ratings to items. Based on these information, the curious recommender agent recommends items to the target user. Formally, the environment can be described by three components $E = \langle User, Item, R \rangle$, where

- $User = \{u, v, \dots\}$ is a set of users,
- $Item = \{i, j, \dots\}$ is a set of items,
- $R(u, i)$ is the observed rating of u for i .

Based on these environmental information, the curious recommender agent will recommend items that both match the target user's preferences and are surprising, so as to stimulate the target user's curiosity. Next, we will present the internal function of a recommender agent that predicts the target user's ratings for unexperienced items.

5.2.2 The Internal Function: Rating Prediction

A basic internal function for any recommender agent is to predict ratings to items that have not been experienced by the target user. A higher predicted rating for an item implies that the item matches better with the target user's preferences. In this work, we adopt the popular Matrix Factorization (MF) method [144] to calculate the predicted rating of a user u to an unexperienced item i , due to its high recommendation accuracy. It should be noted that our model could use a different rating prediction method, if such a method could yield a more accurate prediction. In other words, our model is not strictly dependent on MF.

The basic MF model maps both users and items to a joint latent factor space of dimensionality d , such that user-item interactions are modeled as inner products in that space. Accordingly, each user u is associated with a vector $p_u \in \mathbb{R}^d$, where the elements in p_u measure the user's preferences w.r.t. d features. Each item i is associated with a vector $q_i \in \mathbb{R}^d$, where the elements in q_i measure the item's importance weights for the d features. The *predicted rating score* of user u to item i , denoted by $\hat{R}(u, i)$, is obtained by:

$$\hat{R}(u, i) = p_u^T q_i. \quad (5.1)$$

The values in p_u and q_i are initially assigned arbitrarily and then iteratively updated by a simple gradient decent technique. For each observed rating $R(u, i)$, the latent variable vectors p_u and q_i are updated as follows:

$$p_u \leftarrow p_u + \gamma(\Delta_{ui} \cdot q_i - \lambda \cdot p_u), \quad (5.2)$$

$$q_i \leftarrow q_i + \gamma(\Delta_{ui} \cdot p_u - \lambda \cdot q_i), \quad (5.3)$$

where

$$\Delta_{ui} = R(u, i) - p_u^T q_i. \quad (5.4)$$

Here, γ is the learning rate and λ is a regularization parameter which is used to minimize overfitting until the minimum improvement in accuracy or a predefined number of iterations is reached. To achieve higher accuracy, factors such as bias [151] and trust [67] can be incorporated into the updating rules, but this is beyond the scope of this work.

5.2.3 Curious Functions: Stimulus Detection and Interest Evaluation

The predicted rating score implies the target user's preferences to an item. However, items with high predicted rating scores are often similar to those that the users are familiar with and thus, may be of little interest to them. In order to attract users' interest, the curious recommender agent aims to detect surprising items that may stimulate users' curiosity. In this section, we present the implementation of the two curious functions: stimulus detection and interest evaluation.

5.2.3.1 Stimulus Detection

According to the theory of curiosity in human psychology [7, 145], unexpectedness results in a feeling of surprise, where surprise is one of the key factors that stimulate curiosity. It can be generalized into the social context: a friend's unexpected behaviors may cause a person's feeling of surprise which will then lead to the person's curiosity. Based on this theory, a user u will be surprised if a friend v 's rating to an item i significantly differs from u 's expectation of v 's preference towards i .

The target user's expectation of a friend's preference towards an item can be estimated by a predictive model, e.g., MF, that is trained without the friend's observed rating to this item. To differentiate from the traditional recommender systems where predictions are usually made for items that have no observed ratings, we define *pseudo-predicted rating* of a user v to an item i , if an observed rating $R(v, i)$ exists:

$$\check{R}(v, i) = \check{p}_v^T \check{q}_i, \quad (5.5)$$

where $\check{p}_v$ and $\check{q}_i$ are pseudo-latent vectors trained from all the available user-item ratings except $R(v, i)$. The pseudo-predicted rating $\check{R}(v, i)$ is an estimation of the target user u 's expectation of the friend v 's preference towards item i .

Surprise occurs in two cases: $R(v, i)$ is much larger than $\check{R}(v, i)$ or $R(v, i)$ is much smaller than $\check{R}(v, i)$. The former case represents the phenomenon that a friend gives high ratings to something he/she generally does not like, i.e., positive surprise, whereas the latter case represents the phenomenon that a friend gives low ratings to something he/she generally likes, i.e., negative surprise. In real life, people rarely give importance to the negative surprise during a recommendation task because a person's taste in the field he/she likes is often trusted by his/her friends. For example, Bob likes comedy movies very much and gives a low rating to one comedy movie, then Alice will mostly ignore this movie because she trusts Bob's judgement on such a kind of movies.

Hence, in our recommendation task, we focus on the positive surprise and define the surprise caused by a friend v 's rating to an item i , denoted by $\mathcal{S}(v, i)$, as follows:

$$\mathcal{S}(v, i) = \begin{cases} R(v, i) - \check{R}(v, i), & \text{if } R(v, i) - \check{R}(v, i) > T_E \\ 0, & \text{otherwise.} \end{cases} \quad (5.6)$$

and

$$T_E = \frac{\sum_{u,i} (R(u, i) - \hat{R}(u, i))^+}{N^+} \quad (5.7)$$

where T_E is an error threshold calculated by the mean of all the positive errors. $(R(u, i) - \hat{R}(u, i))^+$ represents ratings that satisfy $R(u, i) > \hat{R}(u, i)$, and N^+ denotes the number of ratings that satisfy $R(u, i) > \hat{R}(u, i)$. In this way, we rule out the surprises caused by the innate errors of MF algorithm.

As MF has been shown in the literature to yield quite accurate predictions, it is expected that an item's pseudo-predicted rating $\check{R}(v, i)$ should be very similar to the observed rating $R(v, i)$. In other words, $\mathcal{S}(v, i)$ should usually be small and close to the error threshold. If $\mathcal{S}(v, i)$ is much larger than the error threshold, it can be due to an unexpected rating behavior of v that deviates from v 's usual rating patterns. For example, for Bob who generally does not like horror movies (always gives low ratings or even rarely watches them), MF tends to predict low ratings for such movies. If one day Bob gives a high rating to a horror movie, then this surprising rating cannot be captured by the previously trained model and results in a high surprise score for this particular movie.

On the social network, one user may have multiple friends. For an item that has not been experienced by the target user, it is possible that multiple friends have given

surprising ratings to it. Hence, to model the target user's curiosity towards this item, there are two issues to be addressed: (1) one user's responses to each individual friend's surprising ratings would be different and (2) how to evaluate curiosity based on the combined surprising effects when multiple friends give surprising ratings to the same item.

To address the first issue, we propose a *surprise correlation* measure between two users who are mutually friends, denoted by $SC(u, v)$, as follows:

$$SC(u, v) = 1 - \frac{\sum_{i \in M(u, v)} \left(\frac{abs(\mathcal{S}(u, i) - \mathcal{S}(v, i))}{R_m} \right)}{|M(u, v)|}, \quad (5.8)$$

where $M(u, v)$ denotes the set of items that both u and v gave surprising ratings to in the history, $||$ is the operator that returns the number of elements in a set, abs is the operator that obtains the absolute value, and R_m is the maximum rating difference for normalization. Surprise correlation reflects the covariance of the historical surprising ratings between two users who are mutually friends, which can be used to predict the target user's responses to a friend's surprising ratings in the future, or whether the target user will also be surprised at the items that surprise the friend.

To address the second issue, we propose three strategies for evaluating the target user's curiosity towards an item i , denoted by the *curiosity score* $\mathcal{C}(u, i)$, as follows:

$$\mathcal{C}(u, i) = \min_{v \in F_s(u)} (SC(u, v) \cdot \mathcal{S}(v, i)), \quad (5.9)$$

$$\mathcal{C}(u, i) = \text{ave}_{v \in F_s(u)} (SC(u, v) \cdot \mathcal{S}(v, i)), \quad (5.10)$$

$$\mathcal{C}(u, i) = \max_{v \in F_s(u)} (SC(u, v) \cdot \mathcal{S}(v, i)), \quad (5.11)$$

where $F_s(u)$ denotes the set of u 's friends who give surprising ratings to item i . Equation (5.9) represents a *conservative* strategy which takes the minimum response aroused by the friends' surprising ratings. Equation (5.10) represents an *average* strategy which takes the average value of the responses to all the friends' surprising ratings. Equation (5.11) represents a *bold* strategy which takes the maximum response aroused by the friends' surprising ratings. The impacts of the three strategies on the recommendation results will be studied in the experiments.

5.2.3.2 Interest Evaluation

A curiosity-stimulating item may not catch a user's attention unless the user is fond of that item to some extent. For example, the fact that Bob gives a high rating to a horror movie may not interest Alice if she also hates horror movies. Hence, we propose an interest score to balance the predicted rating score and the curiosity score for the recommended items.

The *interest score* of a user u for an item i is simply modeled by the sum of the predicted rating score and the curiosity score:

$$I(u, i) = \hat{R}(u, i) + \mathcal{C}(u, i), \quad (5.12)$$

where the curiosity score $\mathcal{C}(u, i)$ acts as a regularization term that adds value to those items that are surprisingly interesting to the target user.

To summarize, the proposed method (with the *bold* strategy) is outlined in Algorithm 2. Line 1 calculates latent vectors for users and items through MF. Line 2 calculates the error threshold for ruling out the surprises caused by the inaccuracy of MF. Lines 4-14 calculate surprise scores for the experienced items that have been rated by a friend. Lines 15-27 calculates interest scores for items that have at least one friend giving a surprising rating to it; else if no friend has given a surprising rating to it, the interest score will be simply given by the predicted rating (Lines 25). This algorithm takes the *bold* strategy to calculate curiosity score as an example (Line 21). If the *average* or *conservative* strategy is to be chosen, just replace Line 21 with Equation (5.10) or Equation (5.9), respectively. Based on this algorithm, the target user can get a recommendation list ranked by the descending order of interest scores.

Let us analyze the computational complexity of the curiosity-driven recommendation algorithm. Let m represents the total number of users and n represents of the total number of items under examination. If MF converges after k rounds, then the computational complexity of the MF training process is kmn . Similarly, the computational complexity of the MF training process for getting the pseudo-predicted ratings for a friend is kmn . The maximum number of friends a user u can get is m and hence the maximum computational complexity for the MF training process is $(m + 1)kmn$, including one MF training process for predicting u 's ratings and m times MF training process for calculating the

Algorithm 2: SOCIAL CURIOSITY DRIVEN RECOMMENDATION ALGORITHM - *bold* STRATEGY

```

1 Run MF to get  $p_u$  and  $q_i$  for each user  $u$  and unexperienced item  $i$ , get  $\hat{p}_v$  and  $\hat{q}_i$ 
  for each friend  $v$  and experienced item  $i$ ;
2  $T_E = \frac{\sum_{u,i} (R(u,i) - \hat{R}(u,i))^+}{N^+}$ ;
3 for each user  $u$  do
4   for each friend  $v \in F(u)$  do
5     for each item  $i$  experienced by  $v$  but not  $u$  do
6        $\hat{R}(v,i) = p_v^T q_i$ ;
7       if  $R(v,i) - \hat{R}(v,i) > T_E$  then
8          $\mathcal{S}(v,i) = R(v,i) - \hat{R}(v,i)$ ;
9       end
10      else
11         $\mathcal{S}(v,i) = 0$ ;
12      end
13    end
14  end
15  for each item  $i$  not experienced by  $u$  do
16     $\check{R}(u,i) = \tilde{p}_u^T \tilde{q}_i$ ;
17    if  $F_s(u) \neq \emptyset$  then
18      for each  $v \in F_s(u)$  do
19         $SC(u,v) = 1 - \frac{\sum_{i \in M(u,v)} (\frac{abs(\mathcal{S}(u,i) - \mathcal{S}(v,i))}{S_m})}{|M(u,v)|}$ ;
20      end
21       $\mathcal{C}(u,i) = \max_{v \in F(u)} SC(u,v) \cdot \mathcal{S}(v,i)$ ;
22       $I(u,i) = \hat{R}(u,i) + \mathcal{C}(u,i)$ ;
23    end
24    else
25       $I(u,i) = \hat{R}(u,i)$ ;
26    end
27  end
28 end
    
```

friends' pseudo-predicted ratings. After training, from Algorithm 2 (Line 3-28), it can be analyzed that the maximum computational complexity for making recommendation is $mn + n$. Hence, the computational complexity of the curiosity-driven recommendation algorithm is polynomial.

5.3 Performance Evaluation of the Curious Recommender Agent

In this section, the proposed methods are evaluated against the benchmark method in terms of accuracy, coverage, and diversity with two large scale real world datasets: Douban and Flixster.

5.3.1 Datasets

Douban is a Chinese website for providing ratings for movies, books and music (rating scale from 1 to 5). We use the Douban movie dataset¹ published in [67]. This dataset contains 129,490 unique users and 58,541 unique movies with 16,830,839 ratings. As to the friend network, the total number of undirected friend links between users is 1,692,952. The statistics of the Douban user-item rating matrix and friend network are summarized in Table 5.1 and Table 5.2, respectively.

Table 5.1: Statistics of User-Item Matrix of Douban.

Statistics	User	Item
Max. Num. of Ratings	6,328	49,504
Ave. Num. of Ratings	129.98	287.51

Table 5.2: Statistics of Friend Network of Douban.

Statistics	Friends per User
Max. Num.	986
Ave. Num.	13.07

Flixster is an English website for providing ratings for movies (rating scale is from 1 to 5). The Flixster dataset² was published in [41]. This dataset contains 1,049,508 unique users and 66,726 movies with 8,196,077 movie ratings. For the friend network, the total number of undirected friend links between users is 7,058,819. We preprocessed the Flixster dataset by removing the large portion of users who have social relations but no expressed ratings because our algorithm is interested in friends' rating behaviors.

¹<https://www.cse.cuhk.edu.hk/irwin.king/pub/data/douban>

²<http://www.cs.sfu.ca/~sja25/personal/datasets/>

After preprocessing, the Flixster dataset contains 147,612 unique users and 66,726 movies with 8,196,077 movie ratings. The total number of undirected friend links between users is 2,538,640. The statistics of the processed Flixster user-item rating matrix and social friend network are summarized in Table 5.3 and Table 5.4, respectively.

Table 5.3: Statistics of User-Item Matrix of Flixster.

Statistics	User	Item
Max. Num. of Ratings	30,977	34,791
Ave. Num. of Ratings	55.52	122.83

Table 5.4: Statistics of Friend Network of Flixster.

Statistics	Friends per User
Max. Num.	579
Ave. Num.	17.20

5.3.2 Metrics

To evaluate the performance of the proposed system, the following popular metrics are adopted:

Coverage: The coverage of a recommender system is a measure of the domain of items in the system over which the system can make recommendations [32]. The most common measure for coverage is the proportion of items that the recommendation system can recommend:

$$Coverage = \frac{|\cup_{u \in U} L(u)|}{N_T} \times 100\% \quad (5.13)$$

where U is the set of users to whom recommendations will be made, $L(u)$ is the set of items recommended to a user u , $\cup$ is the operator that takes the union of sets, and N_T is the total number of items in the recommender system. Coverage value is dependent on the length of the recommendation list $L(u)$, i.e., a longer list often leads to a higher coverage value. To make the comparisons fair, in this experiment, all the results are obtained based on the top 10 recommendations.

Diversity: According to [13], the diversity of a list of items is defined as the average dissimilarity between all pairs of items. The dissimilarity between two items is determined

by the dissimilarity between ratings given to them by the set of users who have rated both. Let $U(i, j)$ represents the set of users who have rated both i and j . The dissimilarity between i and j is given by:

$$dis(i, j) = \frac{\sum_{u \in U(i, j)} \left(\frac{abs(R(u, i) - R(u, j))}{R_m} \right)}{|U(i, j)|}, \quad (5.14)$$

where R_m is the maximum rating difference for normalization.

Therefore, the diversity of a recommendation list $L(u)$ is given by:

$$D(L(u)) = \frac{\sum_{i \in L(u)} \sum_{j \in L(u), i \neq j} (dis(i, j))}{\frac{|L(u)|}{2} (|L(u)| - 1)}, \quad (5.15)$$

The system level diversity is calculated by the average diversity of the recommendation lists for all users:

$$Diversity = \frac{\sum_{u \in U} D(L(u))}{|U|} \quad (5.16)$$

MAE and RMSE: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are both widely adopted accuracy metrics [32]. The definitions for MAE and RMSE are given as follows:

$$MAE = \frac{\sum_{u, i} abs(R(u, i) - \hat{R}(u, i))}{N}, \quad (5.17)$$

and

$$RMSE = \sqrt{\frac{\sum_{u, i} (R(u, i) - \hat{R}(u, i))^2}{N}} \quad (5.18)$$

where $R(u, i)$ is the observed rating of u for i , $\hat{R}(u, i)$ is the predicted rating of u for i , and N is the number of observed ratings. The MAE and RMSE for the proposed social curiosity driven recommendation model are calculated by replacing the predicted rating $\hat{R}(u, i)$ with the interest score $I(u, i)$ in Equation (5.17) and Equation (5.18), respectively.

5.3.3 Methods

The following methods, including the baseline MF method and the three proposed methods, are compared:

1. **MF:** the baseline Matrix Factorization method [144].

2. $\mathbf{SC}_{min}$: the social curiosity driven recommendation model using the *conservative* strategy for curiosity evaluation.
3. $\mathbf{SC}_{ave}$: the social curiosity driven recommendation model using the *average* strategy for curiosity evaluation.
4. $\mathbf{SC}_{max}$: the social curiosity driven recommendation model using the *bold* strategy for curiosity evaluation.

In order to fairly compare these methods, we employ similar parameter settings for those common parameters adopted in them. In this experiment, for the MF parameters in all the four methods, we set the learning rate γ to 0.001, the regularization parameter λ to 0.02, and the latent factor dimension d to 10. Except for these MF parameters adopted for calculating predicted rating score, the rest part of the social curiosity inspired recommendation methods ($\mathbf{SC}_{min}$, $\mathbf{SC}_{ave}$ and $\mathbf{SC}_{max}$) is totally parameter free.

As social curiosity recommendation is closely related to the behaviors of the target user's friends, it is interesting to analyze the impact of the target user's friend number on the recommendation results. In social network research, degree is a common centrality measure that indicates how many people a person can reach directly [53]. With bi-directional friend links, degree refers to the number of the direct friends a person has. We conducted experiments for users from different degree groups: the first group consists of users with degrees from 1 to 20, the second group consists of users with degrees from 20 to 40, and similarly the last group consists of users with degrees from 100 to 120. We do not continue dividing users with degree above 120 into groups because the number of users with degree above 120 takes up less than 1% of the total number of users in the datasets. As coverage value is influenced by the number of users to whom recommendations are made, i.e., a higher number of users leads to a higher level of coverage, and the number of users decreases dramatically in larger degree groups, to make the comparisons fair, we randomly select 1000 users from each degree group for study.

5.3.4 Experiment Results

The experiment results on Coverage, Diversity, MAE and RMSE with Douban dataset and Flixster dataset are shown in Figure 5.2. From the results, we can observe that the

Figure 5.2: Performance Results on Coverage, Diversity, MAE and RMSE with Douban and Flixster Datasets

social curiosity driven recommendation methods (SC_{min} , SC_{ave} and SC_{max}) significantly outperform the baseline MF method in terms of coverage and diversity consistently across all the degree groups, without any significant compensation on accuracy (MAE and RMSE). For Douban dataset, the minimum improvement on coverage is 154.41% (SC_{max}), the minimum improvement on diversity is 1235.75% (SC_{min}), the maximum decrease on MAE is 8.78% (SC_{max}), and the maximum decrease on RMSE is 10.09% (SC_{max}). For Flixster dataset, the minimum improvement on coverage is 37.66% (SC_{max}), the minimum improvement on diversity is 11.15% (SC_{min}), the maximum decrease on MAE is 8.72% (SC_{max}), and the maximum decrease on RMSE is 7.81% (SC_{max}).

By comparing the performances of the three social curiosity driven recommendation methods, i.e., SC_{min} (the conservative strategy), SC_{ave} (the average strategy) and SC_{max} (the bold strategy), it can be observed that the conservative strategy consistently achieves the highest coverage values, whereas the bold strategy consistently achieves the lowest coverage values, across all degree groups for both the Douban dataset and the Flixster

dataset. The reason for this phenomenon can be explained as follows: at the system level, the bold strategy always chooses the most surprising items for recommendation. If the user who gives the highest surprising ratings to these items has a large number of friends, then these items will be repetitively recommended to all his/her friends. Hence, due to the repetitive recommendation of the most surprising items, the coverage achieved by the bold strategy tends to be smaller than that by the conservative strategy. On the contrary, the bold strategy consistently achieves the highest diversity values across all degree groups for both the Douban dataset and the Flixster dataset. This can be due to the reason that the most surprising items tend to be very dissimilar to the popular items and the repetitive recommendation of those most surprising items instills higher diversity into the recommendation lists. With regards to accuracy, the bold strategy always compromises most accuracy whereas the conservative strategy always reserves most accuracy. This is reasonable because surprise is positively related to error in the calculation of accuracy metrics.

Let us analyze the impact of degree on the recommendation results. It can be observed from Figure 5.2(a), Figure 5.2(b), Figure 5.2(d), and Figure 5.2(e) that a larger degree leads to a higher increment in diversity and coverage with all the social curiosity driven recommendation methods (SC_{min} , SC_{ave} , and SC_{max}). However, degree does not have a clear relationship with the recommendation results of the traditional MF method. The reason is that the traditional MF method does not explicitly consider social information during recommendation, which results in no clear trend on coverage and diversity along degree groups. Another interesting phenomenon that can be observed is that the conservative strategy and the average strategy always yield similar results whereas the larger the degree is, the result of the bold strategy diverges farther from the conservative strategy. This is due to the reason that high surprises are rare. The effect of these high surprises will be very much diluted when averaging with the low surprises through the average strategy, leading to a similar result with the conservative strategy. However, the larger the degree is, these rare high surprises get more chances to spread, which magnifies the effects of these high surprises on the recommendation results.

5.4 Summary

In this chapter, we have presented a curiosity-driven recommendation algorithm based on the proposed computational model of curiosity and studied the impact brought by curiosity to recommender agents. The idea of introducing curiosity into recommender agents is inspired by the common observed phenomenon in real life that we are often interested by our friends' abnormal behaviors. This is generally known as social curiosity in psychology. Hence, recommending items that have been given an unexpected rating by a friend may cause the target user's interest. For the curious recommender agent, the computational model of curiosity is implemented based on the popular predictive model by matrix factorization, due to its high recommendation accuracy and efficiency in dealing with large scale user-item rating matrices. The curious recommender agent is carefully studied with two large scale datasets: Douban and Flixster. The experimental results demonstrate that social curiosity significantly enhances recommendation coverage and diversity, while maintaining a sufficient level of accuracy.

To summarize, in Part II of this thesis, we have presented a computational framework of curiosity for intelligent agents, including a generic computational model of curiosity, a curiosity-driven learning algorithm, and a curiosity-driven recommendation algorithm. In the third part of this thesis, we will evaluate the practical values of the computational framework in the domain of virtual worlds for education. In this application domain, we introduce curiosity into two types of pedagogical agents: a virtual peer learner and a virtual learning companion. The virtual peer learner is a background Non-player Character in a virtual learning environment to form a virtual social environment with no direct interactions with the human users, whereas the virtual learning companion directly interacts with human users to enhance their learning experiences.

Part III

Application in Virtual Worlds for Education

Chapter 6

Curious Peer Learner

6.1 Introduction

In Part III of this thesis, we study the impact of computational curiosity in the domain of virtual worlds for educational purposes. This chapter focuses on developing a curious peer learner that realizes the generic computational model of curiosity to enhance believability.

With the advances in computer graphics, communication technologies and networking, virtual worlds are rapidly becoming part of the educational technology landscape [139]. Dede [18] suggests that the immersive interfaces offered by virtual worlds can promote learning, by enabling the design of educational experiences that are challenging or even impossible to duplicate in real world. In recent years, the usage of virtual worlds within the educational context is growing quickly. The New Media Consortium (NMC) Annual Survey on Second Life (SL) received 170% increase in response rate between 2007 and 2008. They also found that many of the educators who earlier used the existing SL, have started creating their own virtual worlds in less than a year's time [30].

Virtual Singapura ¹ is a Virtual Learning Environment (VLE) designed to facilitate the learning of plant transport systems in lower secondary school. It has been employed in various studies, such as design perspectives for learning in VLE, pre-service teachers' perspectives on VLE in science education, productive failure and impact of structure on learning in VLE, slow pedagogy in scenario-based VLE, and what students learn in VLE, etc. [40][55][54][56][133]. Till date, over 500 students in Singapore and over 300 students in Australia have played Virtual Singapura. During the field studies of Virtual Singapura,

¹<http://virtualsingapura.com/game/project/>

several issues with learning in VLE have been observed. First, students tend to spend more time exploring the landscape of the virtual world rather than concentrating on the learning content. Second, some low-functioning students studying alone in VLE often get confused or stuck, and require constant guidance from teachers or game designers to move forward.

Based on these observations, we propose a virtual peer learner to reside in VLE and accompany students to learn. The idea is derived from the common educational practice of peer learning, where students learn with and from each other without the immediate intervention of a teacher [12]. Benefits of a peer learner include: a peer learner can present “learning triggers”, that are interactions or experiences causing students to try new things or to think in novel ways; bi-directional peer relationships can facilitate professional and personal growth; and tapping into a learner’s own experience can be both affirming and motivating [21]. Hence, a virtual peer learner has the potential to engage students and motivate them to spend more time on the learning content. Also, a virtual peer learner can potentially help low-functioning students to think and learn better in VLE.

A key design issue for such virtual characters is their believability [46]: they need to give the users an impression of being lifelike and believable, producing behaviors that appear to the users as natural and appropriate. In order to design a virtual peer learner that can emulate a real student and behave naturally in the learning process, a biologically inspired approach is necessary. In human psychology, studies have shown that curiosity is an important motivation that links cues reflecting novelty and challenge with natural behaviors such as exploration, investigation and learning [52]. In Reiss’s [97] 16 basic desires that motivate our actions and shape our personalities, curiosity is defined as “the need to learn”. Attempts to incorporate curiosity into Artificial Intelligence find curious machines have advanced behaviors in exploration, autonomous development, creativity and adaptation [73] [76] [106] [124]. However, as a basic motivation that drives the learning behaviors of human beings [97], the role of curiosity in a virtual peer learner has not yet been explored. This creates a challenge for introducing curiosity into virtual peer learners and studying its impact on their learning behaviors. In the rest of this chapter, we will tackle this challenge by implementing the proposed computational model of curiosity in virtual peer learners and conducting experimental studies to analyze the effects brought by curiosity.

Figure 6.1: An illustration of the interactions between the curious peer learner and the VLE.

6.2 Curious Peer Learner

In this section, we present how the proposed computational model of curiosity is implemented in virtual peer learners. The high-level interactions between a curious peer learner and the VLE is depicted in Figure 6.1. It shows that the curious peer learner is an agent residing in the VLE. VLE has predefined learning tasks that can be undertaken by the agent to acquire knowledge points. The agent's actions demonstrate its knowledge acquisition strategies (e.g., study, rest, or explore) towards the learning tasks. Next, we will present the implementation of the curious peer learner in detail. We start with the implementation of the environment and basic internal functions, and then discuss the implementation of the curious functions.

6.2.1 Virtual Learning Environment

The goal of the curious peer learner is not to effect changes (of environmental states) in the environment but to acquire the knowledge points embedded in the environment. Hence, instead of focusing on environmental states (as described in Chapter 3), the curious peer

learner pays attention to learning tasks. Hence, the environment is represented by:

$$E = \langle T \rangle \quad (6.1)$$

and

$$T = \{t_1, t_2, \dots, t_n\} \quad (6.2)$$

where T represents a set of learning tasks.

In order to model learning tasks that can be easily interpreted by both the human students and the curious peer learner, we adopt concept maps, a semantic knowledge representation method that stems from the learning theory of constructivism [83]. It has been widely applied in classrooms for knowledge organization [83] and many educational softwares for modeling the mind of students [10, 148].

A concept map is a graph-based representation that describes semantic relationships among concepts. It can be represented by a directed graph with nodes and edges inter-connecting nodes. We formalize the symbolic representation of concept maps as follows:

A concept map CM with n concepts, is defined as $M = \langle C, La \rangle$:

1. $C = \{c_i | c_i \in P_c; i = 1, 2, \dots, n\}$ represents the concepts, where P_c is a set of predefined concepts in VLE;
2. $La = \{la_{ij} | la_{ij} \in P_{la} \cup \{null\}; i = 1, 2, \dots, n; j = 1, 2, \dots, n\}$ represents the labels describing the relationships between two concepts, where P_{la} is a set of predefined labels in VLE.

Based on the above definition, in CM s, concepts and relationships are all semantic expressions. An example is shown in Figure 6.2, wherein the concept set is $\{c_1: \text{water}, c_2: \text{CO}_2, c_3: \text{chloroplast}, c_4: \text{sugar}, c_5: \text{O}_2, c_6: \text{sun light}, c_7: \text{photosynthesis}\}$ and the label set is $\{la_{17}, la_{27}: \text{is the material for}, la_{37}: \text{is the location of}, la_{67}: \text{aid}, la_{74}, la_{75}: \text{produce}\}$.

In the environment E , knowledge in learning tasks are represented by CM s. A knowledge point is defined by a relationship in CM , denoted by $k = (c_i, c_j, la_{ij})$, where $la_{ij} \neq null$. For example, a knowledge point in Figure 6.2 is $(\text{water}, \text{photosynthesis}, \text{is the material for})$. In this way, each learning task $t \in T$ that is designed through CM s, can be represented by a set of knowledge points, denoted by $t = \{k_1, k_2, \dots, k_m\}$.

Figure 6.2: The concept map for the learning task ‘photosynthesis’

For example, the *CM* in Figure 6.2 can be designed to be a learning task with six knowledge points.

The goal of the curious peer learner is to acquire the knowledge points in learning tasks strategically and mimic a human learner’s learning behaviors, so as to create a believable virtual learning environment. Next, we will introduce the basic internal functions of the curious peer learner that support the agent’s knowledge acquisition behaviors.

6.2.2 Internal Functions

This section presents the basic internal functions of the curious peer learner that support the agent’s knowledge acquisition behaviors.

6.2.2.1 Actions

A virtual peer learner’s knowledge acquisition behaviors can be characterized by the actions that it can perform. In order to simulate human-like behaviors, we look at the learning behaviors of a human student. In VLE, a human student can take a variety of actions according to the design of control. Here, we focus on three action categories relating to the acquisition of knowledge: *explore*, *rest* and *study*.

The first category contains actions of a human student to explore interesting learning contents in VLE. Examples include clicking objects, reading information, etc. When a student is tired of learning new stuff, he/she will choose some leisure activities such as roaming around, chatting with friends, etc. These actions are categorized under the group of *rest*. The third category is *study*, including all actions related to gaining knowledge, such as reading, writing, answering questions, etc.

Here, we stay on the level of action categories and do not specify the actions in each category. Hence, actions of the virtual peer learner are represented as follows:

$$A = \{explore, rest, study\} \quad (6.3)$$

where each element in A stands for a category of actions.

6.2.2.2 States

The virtual peer learner's decision-making on action selection is constrained by its inner states. The inner state is defined as follows:

$$S_{in} = \langle energy \rangle \quad (6.4)$$

where *energy* is an intrinsic constraint that the virtual peer learner should take into consideration when choosing actions. This is a natural constraint in any learning environment for a real human student. When a student feels fatigue, he/she will need some rest before continuing to work. Also, different learning tasks may cause different levels of fatigue. For example, browsing through the topics is much easier than studying a topic after it is chosen. Hence, a good student knows how to adjust his/her learning strategy to properly spend energy. The value of *energy* changes as follows:

$$energy = energy + E(a), a \in A \quad (6.5)$$

where E is a function mapping from an action a to its cost of energy. When *energy* < 0 , the virtual peer learner can only choose *rest* to recharge energy. Both the action *explore* and *study* can decrease certain amount of *energy* in its inner state, whereas the action *rest* can recover certain amount of *energy*.

Besides inner states, the virtual peer learner also maintains a set of external states which reflect the agent's relationship with the VLE. As the virtual peer learner's goal is to learn knowledge points in the VLE, the main effort of the agent is to explore the VLE for learning tasks. Hence, the external state is defined as a two tuple:

$$S_{ex} = \langle in_learning_zone, next_to_task \rangle \quad (6.6)$$

where *In_learning_zone* and *next_to_task* are binary values indicating whether the virtual peer learner is in learning zones and whether its location is within a range to learning tasks, respectively. The action *explore* can update *next_to_task* in the agent's external state.

Hence, the state of the virtual peer learner is defined as follows:

$$S = S_{in} \times S_{ex} \quad (6.7)$$

which is the combination of the inner state and the external state.

6.2.2.3 Decision-Making

The extent to which an agent's behavior is similar to a human learner is determined by its decision-making function. In this system, the virtual peer learner only has the knowledge about what actions can be taken in each state, but not the model of the world or the predefined rules on which action to choose. Hence, we adopt a modelless reinforcement learning mechanism, Q-learning [47], for the agent to form decision-making rules dynamically. Q-learning estimates the $Q(s, a)$ values, which are the expected rewards for executing an action a in a state $s \in S$. $Q(s, a)$ values are maintained in a Q-table, based on which the agent will make decisions, i.e., choose the most rewarding actions.

To improve its decision-making ability, the agent can update the Q-table dynamically through reinforcement learning. Each $Q(s, a)$ is updated based on

$$Q(s, a) = (1 - \omega) \cdot Q(s, a) + \omega \cdot (r + \gamma V(s')) \quad (6.8)$$

where

$$V(s') = \max_{a \in A} Q(s', a) \quad (6.9)$$

is the best reward the agent can expect from the new state s' . r is the reinforcement value. γ is the discount factor that defines how much expected future rewards affect decision now. ω is the learning rate that controls how much weight is given to the reward just experienced.

6.2.2.4 Knowledge Acquisition

Knowledge acquisition happens when the agent chooses the action *study* based on its decision-making. Through the action *study*, a certain amount of new knowledge in the current learning task is incorporated into the agent's memory. The part of a learning task t that has been learnt by the virtual peer learner is represented by t_m in the agent's memory.

We define the knowledge acquisition efficiency Δ of a virtual peer learner as :

$$\Delta = l, 0 \leq l \leq 1 \quad (6.10)$$

where l is the base knowledge acquisition ability when the virtual peer learner is in *no_emotion* state. The action *study* is implemented by randomly selecting Δ percentage of new knowledge points in the current learning task and recording them into the agent's memory.

With the internal functions specified above, the agent can form a mechanism of knowledge acquisition. However, it still lacks a mechanism of internal reward generation, which is the important intrinsic drive that regulates the agent's knowledge acquisition strategies. Next, we will introduce the curious functions of the curious peer learner, which generate internal rewards for the agents based on the mechanism of curiosity.

6.2.3 Curious Functions: Stimulus Detection and Interest Evaluation

In human beings, curiosity is the natural motivation for learning, which generates internal rewards to drive human beings to explore the unknown [65]. In order to simulate a similar process in virtual agents, curious functions are designed for the virtual peer learner. Two curious functions are implemented based on the general computational model presented in Chapter 3, including stimulus detection and interest evaluation. The outputs of curious functions form the internal rewards to update the agent's Q-table during reinforcement learning, which regulates its knowledge acquisition strategies.

6.2.3.1 Stimulus Detection

For the curious peer learner, each learning task in VLE is treated as a stimulus. As defined in the previous section, for each learning task $t \in T$, there is a set of knowledge points associated, denoted by $t = \{k_1, k_2, \dots, k_m\}$. This set of knowledge points are intended to be learnt by the agent upon finishing the learning task t .

With concept map based knowledge representation, the most salient factors that can be appraised in a learning task (stimulus) include novelty, conflict and complexity. Novelty and conflict can be reflected by the dissimilarity between knowledge points to be learnt in the learning task (t) and learnt ones in the agent's memory (t_m). Complexity can be reflected by the total amount of knowledge points intended to be learnt in the learning task (t). The appraisal of uncertainty may require more complex knowledge representation that contains uncertain information and the appraisal of surprise may require a mechanism of predicting future outcomes of agent's actions, which will be considered in future works. Hence, for the virtual peer learner, three collative variables are appraised:

$$V = \{\mathcal{N}, \mathcal{F}, \mathcal{P}\} \quad (6.11)$$

where $\mathcal{N}$, $\mathcal{F}$, and $\mathcal{P}$ represent novelty, conflict and complexity, respectively. Next, the appraisal of novelty, conflict and complexity is discussed in detail.

We define a novel knowledge point in t as the knowledge point that is a member of t but does not have a corresponding knowledge point in t_m , with the same order of concepts. This indicates that the agent has not learnt the knowledge point before. All novel knowledge points in t are kept in the novelty set, denoted by $t \dot{-} t_m$. Formally,

$$t \dot{-} t_m = \{k | k \in t \wedge \neg \exists k' \in t_m, c_i = c'_i \wedge c_j = c'_j\}, k = (c_i, c_j, l_{ij}), k' = (c'_i, c'_j, l'_{ij}) \quad (6.12)$$

A conflicting knowledge point in t is defined as the knowledge point that is a member of t and has a corresponding knowledge point in t_m with same order of concepts, but with different labels. This indicates that the agent understands the knowledge point differently from the learning task. All conflicting knowledge points in t are kept in the conflict set, denoted by $t \tilde{-} t_m$. Formally,

$$t \tilde{-} t_m = \{k | k \in t \wedge \exists k' \in t_m, c_i = c'_i \wedge c_j = c'_j \wedge l_{ij} \neq l'_{ij}\}, k = (c_i, c_j, l_{ij}), k' = (c'_i, c'_j, l'_{ij}) \quad (6.13)$$

It can be deduced from the definition that the conflict set operator $\tilde{-}$ is symmetric, i.e. $t\tilde{-}t_m = t_m\tilde{-}t$.

It can also be deduced that set difference $t - t_m$ equals to the union of novelty set and conflict set, i.e. $t - t_m = (t\dot{-}t_m) \cup (t\tilde{-}t_m)$. Hence, the set difference of from t to t_m contains two types of information in this context: novelty and conflict. In order to measure the level of combined novelty and conflict, we extend Tversky's classic set similarity measurement, referred to as the ratio model [135], by introducing asymmetry to the novelty and conflict information contained in the set difference.

According to the ratio model, the similarity between two sets A and B can be represented by [135]:

$$S(A, B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A - B) + \beta f(B - A)}, \alpha, \beta \geq 0 \quad (6.14)$$

where f is a scale function, and α, β define the degree of asymmetry. According to Tversky, f is usually the cardinality of a set, reflecting the salience or prominence of various members in the set. Also, f satisfies additivity, i.e. $f(X \cup Y) = f(X) + f(Y)$. In the ratio model, $S(A, B)$ is interpreted as the degree to which A is similar to B , where A is the subject of comparison and B is the reference. One naturally focuses on the subject of comparison. Hence, the features of the subject are usually weighed more heavily than the features of the reference, i.e. $\alpha > \beta$.

Next, we extend Tversky's ratio model to introduce asymmetric measure to the novelty and conflict subsets in the set difference as follows:

Let

$$g(A - B) = \delta f(A\dot{-}B) + \xi f(A\tilde{-}B), \delta, \xi \geq 0 \quad (6.15)$$

and

$$S(A, B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha g(A - B) + \beta g(B - A)}, \alpha, \beta \geq 0 \quad (6.16)$$

where $g(A - B)$ is a function of the set difference from A to B , with asymmetry introduced to the novelty and conflict subsets. The parameters δ and ξ give importance to novelty and conflict respectively and determine the degree of asymmetry. Thus, $S(A, B)$ measures the similarity between set A and B , with asymmetry between the set difference: $A - B$ and $B - A$ (determined by α and β), as well as asymmetry between the two types of

information contained in the set difference: novelty and conflict (determined by δ and ξ).

$S(A, B)$ gives the measure of similarity between two sets. However, novelty and conflict are contained in the dissimilarity between two sets, as the union of novelty and conflict forms the set difference, i.e. $t - t_m = (t \dot{-} t_m) \cup (t \tilde{-} t_m)$. Hence, in order to measure novelty and conflict, we must define the dissimilarity $D(A, B)$ between two sets:

$$\begin{aligned} D(A, B) &= 1 - S(A, B) \\ &= \frac{\alpha g(A - B) + \beta g(B - A)}{f(A \cap B) + \alpha g(A - B) + \beta g(B - A)}, \alpha, \beta \geq 0 \end{aligned} \quad (6.17)$$

where $D(A, B)$ is the normalized value containing the dissimilarity information between set A and B .

Based on the definition given in (6.17), the difference between knowledge points in task t and agent's memory t_m can be represented by:

$$\begin{aligned} D(t, t_m) &= 1 - S(t, t_m) \\ &= \frac{\alpha g(t - t_m) + \beta g(t_m - t)}{f(t \cap t_m) + \alpha g(t - t_m) + \beta g(t_m - t)}, \alpha, \beta \geq 0. \end{aligned} \quad (6.18)$$

In the appraisal of curiosity, t is the subject of comparison and t_m is the reference. Here, we give full importance to the subject t , because only the difference from t to t_m , i.e. $t - t_m$, reflects the stimulus's information, consisting of novelty and conflict. The difference from t_m to t , i.e. $t_m - t$, also contains two sources of information: (1) learnt knowledge points that are not given in the learning task, i.e. $t_m \dot{-} t$, and (2) conflicting knowledge points, i.e. $t_m \tilde{-} t$. However, $t_m \dot{-} t$ does not reflect the stimulus's property but rather the agent's knowledge not given in task t . Also, $t_m \tilde{-} t$ has been considered in $t - t_m$ (due to the symmetry of operator $\tilde{-}$). Hence, in the appraisal of curiosity, we assign $\alpha = 1$ and $\beta = 0$. As a result, the difference between t and t_m can be simplified as:

$$\begin{aligned} \hat{D}(t, t_m) &= 1 - S(t, t_m) \\ &= \frac{g(t - t_m)}{f(t \cap t_m) + g(t - t_m)} \\ &= \frac{\delta f(t \dot{-} t_m) + \xi f(t \tilde{-} t_m)}{f(t \cap t_m) + \delta f(t \dot{-} t_m) + \xi f(t \tilde{-} t_m)}, \delta, \xi \geq 0. \end{aligned} \quad (6.19)$$

Hence, the combined appraisal value of novelty and conflict regarding a learning task t can be represented as follows:

$$v(\mathcal{N}, \mathcal{F}, t) = \hat{D}(t, t_m) \quad (6.20)$$

Now, let us consider the third factor that governs the stimulus selection- complexity. In the context of VLE, complexity of a task t can be measured by the normalized salience of all knowledge points contained in the task, represented by:

$$v(\mathcal{P}, t) = \frac{f(t)}{\max_{t' \in T} f(t')} \quad (6.21)$$

where $v(\mathcal{P}, t)$ represent the appraisal value of complexity regarding learning task t .

Here, we model complexity as a scaling factor for $v(\mathcal{N}, \mathcal{F}, t)$, because the value of novelty and conflict can be amplified in very complex tasks and reduced in very simple tasks. For example, searching for an intended piece in a jigsaw puzzle with 1000 pieces is more difficult than searching in one with 10 pieces. Hence, the final stimulation level of a learning task t is defined as:

$$v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) = v(\mathcal{P}, t) \cdot v(\mathcal{N}, \mathcal{F}, t) \quad (6.22)$$

where $v(\mathcal{P}, t)$ is the appraisal of complexity and $v(\mathcal{N}, \mathcal{F}, t)$ is the combined appraisal of novelty and conflict in a stimulus as given in (6.19).

6.2.3.2 Interest Evaluation

The existence of a stimuli does not necessarily results in curiosity unless its stimulation level is optimal. According to Wundt (Figure 6.3), there is an inverted “U-shape” relationship between the stimulation intensity and arousal of emotions [149]. Three emotions are associated along the spectrum of stimulus intensity, where too little stimulation results in boredom, too much stimulation results in anxiety, and optimal stimulation results in curiosity. The curiosity-related emotions are denoted as follows:

$$Em = \{\mathcal{C}, \mathcal{B}, \mathcal{A}\} \quad (6.23)$$

where $\mathcal{C}$, $\mathcal{B}$, and $\mathcal{A}$ represent curiosity, boredom, and anxiety, respectively.

Figure 6.3: The Wundt curve

Based on Wundt's theory, the appraisal of curiosity-related emotions is modeled as follows:

$$\begin{aligned}
 &\text{If } v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) \leq \theta_1 \Rightarrow \mathcal{B}, \\
 &\text{If } \theta_1 < v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) < \theta_2 \Rightarrow \mathcal{C}, \quad 0 \leq \theta_1 \leq \theta_2 \leq 1 \\
 &\text{If } v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) \geq \theta_2 \Rightarrow \mathcal{A}.
 \end{aligned} \tag{6.24}$$

where $v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t)$ is the stimulation level of learning task t , obtained from (6.22), and θ_1, θ_2 are thresholds that split the stimulus intensity axis into three emotion regions. The two thresholds determine the curious peer learner's propensity towards each emotion. For example, if θ_1 is close to 0 and θ_2 is close to 1, then the virtual peer learner will become easily curious about any learning task. On the contrary, if θ_1 is very close to θ_2 , then the virtual peer learner will have a narrow curious region and become very picky about learning tasks.

6.2.3.3 Roles of the Curiosity-Related Emotions

Emotions perform vital functions in human decision-making. Psychological studies show that humans voluntarily seek novel things due to the pleasure of satiating curiosity [8]. Artificial Intelligence research frequently assumes that the human decision-making process consists of maximizing positive emotions and minimizing negative emotions [5, 100, 103]. Based on these observations, curiosity-related emotions are employed as reinforcement functions to guide the virtual peer learner's learning of behavior strategies. The emotion *curiosity* gives positive reinforcement, while both *boredom* and *anxiety* give negative reinforcement.

Figure 6.4: The architecture of the curious peer learner

Also, in humans, emotion can influence action strengths [33]. For example, a student who is interested in a subject will concentrate more and achieve higher knowledge acquisition efficiency than one who is bored with the same subject. Hence, the second role of curiosity-related emotions is to influence actions. Here, we consider their influence on action *study*. The virtual peer learner's knowledge acquisition efficiency Δ can be influenced by emotions as follows:

Based on all the previous discussions, an architecture of the curious peer learner is summarized in Figure 6.4. It can be observed that the curious peer learner senses external states (e.g. in a learning zone) and receives external stimuli (e.g. learning tasks). The external stimuli can trigger the curious peer learner to perform curiosity appraisal. The curiosity appraisal requires learnt knowledge stored in the agent's memory and consists of two functions: stimulus detection and interest evaluation. Emotions derived from the curiosity appraisal process act two roles: (1) as reinforcement value for the learning of state-action mapping, and (2) as influence on action strengths (e.g. the depth of learning). Actions (e.g. explore) derived from the learning of state-action mapping module are performed by actuators, and update intrinsic constraints of the agent (e.g. energy).

$$\Delta = l + F(e), e \in Em \quad (6.25)$$

where F is a mapping from the emotion to its influence on the agent's knowledge acquisition efficiency. Here, $F(\mathcal{C})$ returns a positive value, while $F(\mathcal{B})$ and $F(\mathcal{A})$ both return negative values. Δ is always capped between 0 and 1.

Algorithm 3: PSEUDOCODE FOR THE CURIOUS PEER LEARNER

```

1 while true do // Agent running
2   Perceive current state  $S$ ;
3   Look up Q-table and select the most rewarding action  $a$ ;
4   if  $a$  is study then // Curiosity Appraisal
5     Perceive the current learning task  $t$ ;
6     Calculate novelty and conflict through  $v(\mathcal{N}, \mathcal{F}, t) = \hat{D}(t, t_m)$ ;
7     Calculate complexity through  $v(\mathcal{P}, t) = \frac{f(t)}{\max_{t' \in T} f(t')}$ ;
8     Calculate combined curiosity stimulation through
        $v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) = v(\mathcal{P}, t) \cdot v(\mathcal{N}, \mathcal{F}, t)$ ;
9     if  $v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) \leq \theta_1$  then // Boredom
10      Generate negative rewards and update Q-table;
11      Update agent's knowledge in memory with a low learning efficiency;
12    end
13    else if  $\theta_1 \leq v(\mathcal{N}, \mathcal{F}, \mathcal{P}, t) \leq \theta_2$  then // Curiosity
14      Generate positive rewards and update Q-table;
15      Update agent's knowledge in memory with a high learning efficiency;
16    end
17    else // Anxiety
18      Generate negative rewards and update Q-table;
19      Update agent's knowledge in memory with a low learning efficiency;
20    end
21    Update energy  $energy = energy + E(study)$ ;
22  end
23  else if  $a$  is explore then // Look for learning tasks
24    Move one step in a random direction on the grid;
25    Update energy  $energy = energy + E(explore)$ ;
26  end
27  else // Rest
28    Update energy  $energy = energy + E(rest)$ ;
29  end
30 end

```

The pseudocode for the curious peer learner is given in Algorithm 3. Line 1-3 are to determine the agent’s actions based on the currently perceived world state and the agent’s Q-table. Line 4-22 are to execute the *study* action, which involves the two curiosity appraisal processes: stimulus selection (Line 6-8) and interest evaluation (Line 9-20). During the *study* action, the agent also updates its knowledge in memory and Q-table based on the rewards generated by the curiosity appraisal process. Finally, the agent updates its internal energy state. Line 23-25 are to execute the *explore* action, which move the agent’s position on the grid one step further in a random direction and consumes certain amount of energy. Line 27-29 are to execute the *rest* action, which resumes certain amount of energy. From Algorithm 3, it can be observed that except for the while loop which keeps the agent running, the algorithm does not involve any other loops. Hence, the algorithm is very computationally efficient.

6.3 Performance Evaluation of the Curious Peer Learner

In order to study the impact of curiosity on virtual peer learners’ behaviors, we simulate virtual peer learners in VS and conduct two sets of experiments. The first set of experiment compares the learning behaviors of a curious peer learner with what we expect from the human students. The second set of experiment compares the learning efficiency between a curious peer learner and a non-curious peer learner. Next, we will present the experiments in detail and the experimental results obtained.

6.3.1 Virtual Singapura

Virtual Singapura (VS) is a virtual learning environment designed for lower secondary school students to learn the plant transport systems. The virtual world environment in VS is illustrated in Figure 6.5, where Figure 6.5(a) shows the landscape of VS, designed based on 19th century Singapore. Figure 6.5(b) shows the ant hole trigger where students can shrink their avatars to go inside the tree environment.

For the proposed curious peer learner, we follow the methodology of Learner-Centered Design [127]. It advocates that technology applications must focus on the needs, skills, and interests of the learner. Before developing the curious peer learner, we had described

Figure 6.5: Screen shots of the virtual environment in VS

Figure 6.6: An illustration of the interaction scenarios for a curious peer learner in VS

the functionalities of a curious peer learner and illustrated possible interaction scenarios. Some examples are shown in Figure 6.6. It can be observed from Figure 6.6(a) that the curious peer learner (embodied by a butterfly) is posing questions to a student in order to stimulate his/her thinking on the learning content. Figure 6.6(b) demonstrates that the curious peer learner is directing the student's attention to the potentially interesting learning content.

We conducted a pre-study among several students who have played VS. Responses from questionnaires and interviews have shown that students are interested in having a curious peer learner in VS. Some of the comments from the students include: "I prefer to have a virtual friend to play with me", "I will have a better knowledge if I can compare my knowledge with a virtual friend", "A virtual friend who can be curious about the ant holes, water molecules, and shooting games will make me feel more interested".

6.3.2 Experiment Setup

In order to test the effectiveness of curiosity on the virtual peer learner's adaptation of behavior strategy, we build up a simulation environment in VS. The simulation environment consists of two main elements:

- Learning Zones (LZs) that contain predefined learning tasks.
- Virtual peer learners that reside in VLE to take learning tasks.

6.3.2.1 Generation of Learning Tasks

For each LZ, we generate learning tasks based on one expert CM with 20 concepts and 200 relationships. From this expert CM, we spawn 15500 learning tasks. Each learning task is a submap of the expert CM with 5 concepts randomly chosen. Hence, in one LZ, there are in total 200 knowledge points and 15500 learning tasks. Some of the knowledge points will be repeated in different learning tasks and the agent can not learn more than the number of predefined knowledge points (200) in one LZ.

6.3.2.2 Parameter Setting for the virtual peer learners

In this experiment, we focus on examining the effect of curiosity-related emotions by comparing the performance of virtual peer learners with and without curiosity-related emotions. Hence, we simulate two types of virtual peer learners: a curious peer learner with curious functions and a non-curious peer learner without curious functions. The parameter setting for the two types of virtual peer learners is summarized in Table 6.1.

Firstly, there are four parameters ($\delta, \xi, \theta_1, \theta_2$) regarding the curious functions. Only the curious peer learner will have curious functions. Hence, the four parameters can

Table 6.1: The parameter setting for the curious peer learner and the non-curious peer learner.

Functionality Related	Parameter	Value
Curiosity Appraisal (only for curious peer learner)	δ	1
	ξ	1
	θ_1	0.3
	θ_2	0.8
Reinforcement Learning	ω	0.5
	γ	0.8
Knowledge Acquisition	l	0.5

be understood as describing a specific personality towards curiosity. As this work only considers the comparison between agents with and without curiosity, the parameters are chosen to represent a curious peer learner with an intermediate level of curiosity. δ and ξ are non negative real numbers that give importance to novelty and conflict respectively (Equation 6.19). If δ is greater than ξ , then the agent will focus more on novelty. This means that the agent will magnify the contribution of novelty to stimulation level and lessen the contribution of conflict to stimulation level. Here, we consider the curious peer learner with equal preference to novelty and conflict, and set δ and ξ both as 1. For the two parameters that split the stimulus intensity axis (Figure 2.1) into three regions, we choose intermediate level for both θ_1 and θ_2 (Equation 6.24), with $\theta_1 = 0.3$ and $\theta_2 = 0.8$. This is intuitive because most of humans will have an intermediate level of appraisal on curiosity-related emotions, and few people will have extremes such as no region for curiosity ($\theta_1 = \theta_2$) or no region for negative emotions ($\theta_1 = 0$ and $\theta_2 = 1$).

Secondly, both curious and non-curious peer learners will perform reinforcement learning with the same parameter settings. The difference is that the curious peer learner has curiosity-related emotions as intrinsic rewards to update its Q-table, whereas the non-curious peer learner's Q-table is static because it has no emotions to regulate the learning strategies. ω and γ are the two parameters determining the reinforcement learning process in Equation 6.8. They are all real numbers within the range of $[0,1]$. ω is the learning rate that controls how much weight is given to the reward just experienced. A high value of ω can cause very sudden changes in the learnt Q-value, while a very low value of this parameter causes the learning process to be slow. Hence, an intermediate value of ω is

favorable. In this experiment, we set $\omega = 0.5$. The parameter γ is the discount factor that defines how much expected future rewards affect the decision now. A high value of this parameter gives more importance to future rewards, while a low value gives more importance to current rewards. From the experiment, it showed that the agent needs a comparatively high value of γ to learn a proper behavior strategy. Hence, we set γ with the value 0.8.

Thirdly, the base knowledge acquisition efficiency l in Equation 6.10 determines the knowledge acquisition ability of virtual peer learners. l is a real number within the range of $[0,1]$. The higher the value, the higher percentage of new knowledge will be accurately acquired and integrated into the agent's memory. We set the same base knowledge acquisition efficiency for both the curious and non-curious peer learners. The difference is that the curious peer learner's knowledge acquisition efficiency is influenced by the curiosity-related emotions, whereas the non-curious learner's knowledge acquisition efficiency is static because it has no emotions. In this experiment, we do not consider virtual peer learners with extremely high learning abilities or extremely low learning abilities. For example, a virtual peer learner with $l = 1$ can learn everything in a learning task and a virtual peer learner with $l = 0$ can learn nothing in a learning task. Hence, we choose to simulate virtual peer learners with an intermediate level of learning ability, and set l as 0.5.

6.3.2.3 Function Mappings

The function mappings are summarized in Table 6.2. Function $f(A)$ is the scaling function in Equation 6.19, reflecting the salience of various members in a set. Here, we adopt the most commonly used scale function in Tversky's ratio model, i.e. the cardinality of a set [135].

The energy cost function in Equation 6.5 determines the intrinsic constraint of the virtual peer learners. In this system, the function acts as a system rule that both the curious and non-curious peer learners should consider while choosing action strategies. Here, the settings are purely determined by the system designers. We set the output of actions as follows: *rest* recharges energy fully to 10, *explore* consumes a small amount of energy by 0.05 and *study* consumes a large amount of energy by 3. This is because

Table 6.2: The specification of function mapping for the curious peer learner.

Function	Input	Output
$f(A)$	A	$ A $
$E(a)$	<i>rest</i>	recharge fully to 10
	<i>explore</i>	-0.05
	<i>study</i>	-3
$F(emotion)$	$\mathcal{B}$	-0.2
	$\mathcal{C}$	+0.3
	$\mathcal{A}$	-0.4

browsing through topics (explore) will be much easier than starting to work on the topic (study) after it is chosen.

Similarly, for the function mapping from emotions to their influence on the agent’s knowledge acquisition efficiency in Equation 6.25, we set the influence as -0.2 for $\mathcal{B}$, +0.3 for $\mathcal{C}$ and -0.4 for $\mathcal{A}$. This is based on human’s natural tendency. A student with positive emotions tends to achieve better knowledge acquisition efficiency due to mental excitement. On the contrary, the knowledge acquisition efficiency of a student with negative emotions can be harmed due to the intrinsic resistance to learning.

6.3.3 Experiment Results

In this section, two sets of experiment are conducted and the results are analyzed.

6.3.3.1 The Effect of Curiosity on Learning Behaviors

In this experiment, we analyze the effect of curiosity on the learning of state-action mapping in the curious peer learner. We split the simulation into two phases: reinforcement learning phase and steady phase. In the reinforcement learning phase, the curious peer learner is allowed to learn a behavior strategy based on random exploration of state-action pairs and the intrinsic rewards generated by curiosity-related emotions. In the steady phase, the curious peer learner is put into a new LZ to exploit the behavior strategy learnt in the reinforcement learning phase. For each phase, we ran 10^4 steps to allow sufficient time for Q-table to converge.

In Figure 6.7, we plot the agent’s behavior in both the reinforcement learning and steady phases. The x-axis shows the simulation steps, where the left half is the rein-

Figure 6.7: The curious peer learner’s learning behavior in training and testing phases

forcement learning phase and the right half is the steady phase. Figure 6.7(a) shows the number of knowledge points learnt in each simulation step when the action *study* is taken. It can be observed that the activation pattern in Figure 6.7(a) corresponds to the activation pattern of *study* in Figure 6.7(b). Figure 6.7(b)(c)(d) plot the activation pattern for action *study*, *explore* and *rest*, respectively. It can be observed from Figure 6.7 that the curious peer learner learnt a strategy in the reinforcement learning phase. This is shown by the significantly different behavior patterns in the two phases.

Let us consider Figure 6.7(b) ‘STUDY’. It can be seen that the curious peer learner took a significantly less number of action *study* in the steady phase than in the reinforcement phase. Also, the number of knowledge points learnt (shown in Figure 6.7(a) ‘KNOWLEDGE POINT LEARNT’) in the steady phase maintained a comparatively higher level than that in the reinforcement learning phase. It can be deduced that the curious peer learner learnt an ‘intelligent’ strategy similar to what we expect from human students: they will take learning tasks only when interested and learn with a high efficiency, rather than take whichever learning task appearing in front of them.

Next, let’s analyze Figure 6.7(c) and Figure 6.7(d). It can be observed that in the reinforcement learning phase, the curious peer learner randomly chose *explore* or *rest*

with no strategy. But in the steady phase, the curious peer learner would continuously do *exploration* before an interesting learning task was detected, on the condition that the agent had enough energy. This resembles the behavior of human students: in learning process, they will first search for topics, and only when an interesting topic is found, they will start working on it. When they feel tired, they will take a rest before continuing search or study.

Hence, it can be deduced that curiosity-related emotions successfully guide the curious peer learner to learn a natural state-action mapping. The positive rewards from $\mathcal{C}$ and the negative rewards from $\mathcal{B}$ and $\mathcal{A}$ can guide the curious peer learner to search for interesting learning tasks with intermediate level of stimulation and avoid learning tasks with either too low or too high stimulation. This phenomenon is in line with Wundt's theory (Figure 6.3).

6.3.3.2 The Comparison of Exploration Breadth and Depth between a Curious Peer Learner and a Non-Curious Peer Learner

In this experiment, we compare the performance of two types of virtual peer learners: a curious peer learner with curious functions and a non-curious peer learner without curious functions. In order to analyze the performance of the two virtual peer learners, two indicators are defined:

1. Exploration breadth: This indicator is measured by the number of learning tasks browsed by the agent in a period of time. It indicates the virtual peer learner's tendency to explore for interesting learning tasks.
2. Exploration depth: This indicator is measured by the average number of knowledge points learnt per learning task. It indicates the virtual peer learner's average knowledge acquisition efficiency.

In this experiment, we first trained the two virtual peer learners in one LZ. Then, we put them in another LZ for testing. We run 10^4 steps for both training and testing phase. In order to compare the performance of the two virtual peer learners, they always perform in same LZs. The behavior of the two virtual peer learners in the testing phase are shown in Figure 6.8 and Figure 6.9, respectively.

Figure 6.8: The curious peer learner's learning behavior in the testing phase

Figure 6.9: The non-curious peer learner's learning behavior in the testing phase

By comparing Figure 6.8 and Figure 6.9, it can be observed that the curious peer learner learnt a behavior strategy resembling that of a human student, while the non-

curious peer learner chose actions randomly and behaved irrationally. The curious peer learner followed a behavior pattern that shows continuity in exploration for interesting learning tasks. It only studied learning tasks when its curiosity was aroused and tried to minimize negative emotions by avoiding tasks that are too low or too high in stimulation. The tendency of maximizing positive emotions guarantees an averagely high knowledge acquisition efficiency. However, the non-curious peer learner randomly chose to explore or rest without showing a continuity in the exploration behavior. Also, it did not choose learning tasks deliberately and spent a lot of time on learning tasks that were not interesting and learnt with very low efficiency.

The exploration breadth and depth of both virtual peer learners are summarized in Table 6.3. The exploration breadth is calculated by the total number of learning tasks browsed by the agent in the testing phase. It can be observed that the curious peer learner explored 949 tasks while the non-curious peer learner explored only 386 tasks. The exploration breadth of the curious peer learner is around 3 times that of the non-curious peer learner. The exploration depth is calculated by the average number of knowledge points learnt in the learning tasks studied by the agent in the testing phase. It can be seen that the curious peer learner learnt 5.33 (128 knowledge points learnt in total and studied 24 learning tasks) knowledge points per learning task, while the non-curious peer learner only learnt 0.74 (138 knowledge points learnt in total and studied 187 learning tasks) knowledge points per learning task. The number 0.74 is less than 1 because for the non-curious peer learner, it sometimes learnt nothing new in the learning tasks taken. This indicates that for the non-curious peer learner, it wasted a lot of time on studying learning tasks that are not interesting and gained little knowledge from them.

Table 6.3: The comparison of exploration breath and exploration depth between the virtual peer learners with curiosity and without curiosity.

Measure	Curiosity	No Curiosity
Exploration breadth	949	386
Exploration depth	5.33	0.74

Hence, it can be deduced that curiosity-related emotions can directly guide the virtual peer learner to learn strategic behavior resembling a natural student and indirectly influence the acquisition of knowledge. The curiosity-related emotions drive the virtual

peer learner to enhance both its exploration breadth and depth. This mimics real life where a curious student tends to explore more as well as learn deeper than a student who is less curious.

6.4 Summary

In this chapter, we presented a curious peer learner. It is motivated by the theory that curiosity drives the exploratory behaviors in human beings, such as learning and investigation. Hence, modeling curiosity in virtual peer learners may allow them to demonstrate human-like learning strategies in a virtual learning environment. The curious peer learner is proposed based on the generic computational model of curiosity. The computational model of curiosity is implemented based on the computational theories of concept maps and reinforcement learning. Concept maps allows the knowledge representations to be easily interpreted by both the human learners and the virtual agents. The emotion states derived from the curious functions generate intrinsic rewards to update the agent's Q-table and regulate its learning strategies. Experiment studies through computer simulations show positive impact of curiosity on the performance of virtual peer learners. Simulation results demonstrate that curiosity-related emotions can guide the virtual peer learner to behave as naturally as a real student does. Also, the comparison between a curious peer learner and a non-curious peer learner show that the curious peer learner can demonstrate a higher tendency for exploration (exploration breadth), as well as a higher average knowledge acquisition efficiency (exploration depth).

The rationale behind this research is that a virtual peer learner may have the potential to practise "peer learning", a common educational practice that can benefit learning in multiple aspects. A virtual peer learner residing in VLE can possibly engage students and motivate them to spend more time on the learning content. Also, a virtual peer learner can potentially help low-functioning students to think and learn better in VLE. The virtual peer learner presented in this chapter is a background character in the VLE to form a virtual social environment for learning, which does not directly interact with the human users. However, many educational practices are instantiated through the interactions between teachers and students, or between peer learners [15, 137]. Hence, it

would be beneficial if the virtual characters can directly interact with the human students and provide learning scaffolds when necessary. In the next chapter, we will present the implementation of the computational model of curiosity in another type of pedagogical agent: a curious learning companion, which directly interacts with the human students to enhance their learning experiences.

Chapter 7

Curious Learning Companion

7.1 Introduction

In the previous chapter, we presented a virtual peer learner which realizes the generic computational framework of curiosity. Virtual peer learners are background characters to enrich the virtual learning environment, which do not directly interact with users. However, many educational practices are instantiated through the interactions between teachers and students or between peer learners [15, 137]. Hence, a virtual learning companion that can interact with the users would be interesting to enhance their learning experiences. In this chapter, we focus on developing a virtual learning companion which realizes the generic computational framework of curiosity to provide more meaningful interactions with the users.

Teaching and learning are highly social activities [57]. With the goal to bring a social context into Virtual Learning Environment (VLE), a growing interest has been shown in designing virtual learning companions that adopt a peer metaphor to simulate peer interactions. Virtual learning companions allow human learners to take advantage of the cognitive and affective gains of human peer-mediated learning in a VLE [57]. The general role of a virtual learning companion is illustrated in Figure 7.1. It can be shown from this figure that a human learner is the main actor in a VLE, who acts upon the environment and learns from it. A virtual learning companion performs as a peer who observes the human learner's actions and their effects, i.e., the environmental changes. Based on the observations, the virtual learning companion performs cognitive and affective reasoning to provide appropriate peer-like interactions with the human learner.

Figure 7.1: An illustration of the role of a virtual learning companion in VLE.

Curiosity is an emotional motivation related to exploratory behaviors such as learning, investigation, and exploration, which drives human beings to ask questions and explore for answers. According to the information-gap theory by Loewenstein [65], curiosity can be viewed as arising when attention becomes focused on a gap in one's knowledge. Hence, modeling human-like curiosity in a virtual learning companion may allow the companion to discover knowledge gaps and ask relevant questions. These questions can add new ingredients into the interactions provided by the companion, which may help human learners notice the weakness in their knowledge structure and motivate them to actively explore the VLE. However, curiosity has not been studied as a key personality trait in existing virtual learning companions, which creates a challenge for introducing this novel characteristic into such agents and studying the impact brought by it. In the rest of this chapter, we will tackle this challenge by implementing the generic computational framework of curiosity in virtual learning companions and conducting field studies with human users to analyze how an agent with curiosity will impact the human users' learning experience in a human-agent interaction context.

7.2 Curious Learning Companion

The goal of a curious learning companion is to detect users' knowledge gaps through curiosity appraisal and direct their attention to these knowledge gaps through thought-provoking questions. In this section, we will present the implementation of the curious

learning companion in detail. We start with the representation of the environment, and then introduce the basic internal functions that allow the agent to construct a world model. After that, we introduce the curious functions that are at the core of detecting users' knowledge gaps.

7.2.1 Virtual Learning Environment

VLE is the place where both the human learner (user) and the curious learning companion reside. The human learner is the main performer who acts upon the VLE to achieve learning goals. The curious learning companion does not directly act upon the VLE but accompanies the human learner and provides peer-like interactions to facilitate he/she achieving the learning goals. For the curious learning companion, the VLE is described by five components $E = \langle O, F, A^E, S, K, Evt \rangle$, where

- $O = \{o_1, o_2, \dots, o_n\}$ is a set of objects,
- $F = \{f_1, f_2, \dots, f_m\}$ is a set of facts,
- $S = \{s_1, s_2, \dots, s_l\}$ is a set of world states,
- $A^E = \{a_1, a_2, \dots, a_p\}$ is a set of actions that are allowed to be performed by the human learner on E ,
- $K = \{k_1, k_2, \dots, k_q\}$ is a set of knowledge points,
- $Evt = \{evt_s, evt_k\}$ denotes two types of events, where
 - evt_s happens when the human learner performs an action $a \in A^E$ on E , leading to environmental changes.
 - evt_k happens when the human learner detects a knowledge point k .

In the VLE, each learning concept is visualized as an object o and each fact f is a proposition describing an object. The object described by a fact f is denoted by $obj(f)$. For example, the concept 'car' can be embodied by a car object $o:car$ and a fact about the car can be $f:at\ the\ garage$. Here, $obj(f:at\ the\ garage) = o:car$. The current state $s_i \subseteq F$ contains all the true facts and s_{i+1} is the updated state of s_i .

The ability of a VLE to change is constrained by the set of actions that are allowed to be performed on it. In E , each action $a \in A^E$ has precondition $pre(a) \subseteq F$. a is applicable when $pre(a) \subseteq s_i$. The effect of an action $eff(a)$ is composed of two sets of facts: $add(a) \subseteq F$ and $del(a) \subseteq F$. When a is applied, the facts in $add(a)$ are added to the world state and the facts in $del(a)$ are removed from the world state, i.e., $s_{i+1} = s_i \cup add(a) \setminus del(a)$. The object that action a is performed on is denoted by $obj(a)$. For example, the action $a:drive$ is performed on $o:car$, i.e., $obj(a:drive) = o:car$. A precondition of $a:drive$ is $f:key \text{ is available}$ and an effect of $a:drive$ is $f:car \text{ not at the garage}$. Note that the actions are described as part of the environment because they are not performed by the agent but by the human learners. As mentioned earlier, the agent's goal is not to effect changes in the environment but to observe the human learners' behaviors and detect their knowledge gaps with the environment.

It can be inferred that the effect of certain actions can form the precondition of other actions. In the VLE, a learning goal is achieved by the correct execution of a sequence of actions. For example, to achieve the goal of *driving to the supermarket*, action $a:take \text{ key}$ whose effect is $f:key \text{ is available}$ should be performed before $a:drive$. To help the human learner form the correct plan of actions, the virtual world E provides knowledge points K that can be learnt by the human learner. Each $k \in K$ describes the relationship between a fact and an action, denoted by $k : R(f, a) \in \{r^{pre}, r^{add}, r^{del}, r^{nil}\}$, where r^{pre} means $f \in pre(a)$, r^{add} means $f \in add(a)$, r^{del} means $f \in del(a)$, and r^{nil} means f has no relationship with a . For example, $R(f:key \text{ is available}, a:take \text{ key}) = r^{add}$ and $R(f:key \text{ is available}, a:drive) = r^{pre}$. A knowledge point can be implemented as a piece of message that is delivered by text or Non-Player Characters (NPCs) such as a teacher agent.

There are two types of events in E : evt_s and evt_k . evt_s happens when the world state changes. For example, the garage becomes empty because someone drives the car away. evt_k happens when a new knowledge point is discovered. For example, a neighbor informs that the main road is temporarily shut down due to construction. These events will trigger different agent functions, which will be introduced in the following subsections.

In the VLE described above, the human learners need to accomplish learning goals by forming a series of plans. If their plan fails, they need to discover knowledge points in the VLE and revise the plan till the learning goals are accomplished. In this way, the

human learners gain knowledge through a pattern of try-and-error. However, extensive try-and-error often discourages a person's motivation to learn. Hence, the curious learning companion's goal is to detect the human learners' knowledge gaps, and direct their attention towards these knowledge gaps through thought-provoking questions, which will help to stimulate their motivation to learn. Next, we will present the basic internal functions of the agent that form a model of the world and keep the model updated.

7.2.2 Internal Functions: Planning and Learning

In order to detect a user's knowledge gaps, the agent should maintain a model of the world that is similar to the user's current one. To form such a world model, a plan-based approach is adopted in this work due to its popularity in a number of complex real-time autonomous systems [28]. The effectiveness of explicit representations of plans enables the easy reasoning of future outcomes. As the user is always learning in the VLE, the agent should be able to update its world model when the user gains new knowledge. Hence, a learning mechanism is necessary for the agent to keep its world model updated to the user's current knowledge level. Next, we will present the two basic internal functions of the curious learning companion, i.e., planning and learning, in detail.

7.2.2.1 Planning

The completeness of the world model depends on the set of knowledge points possessed by the agent. The agent maintains a set of knowledge points in its memory, which is denoted by $M = \{k_1, k_2, \dots, k_i\}$, where $M \subseteq K$. These knowledge points can be transformed into a set of plans $Plan = \{plan_1, plan_2, \dots, plan_j\}$, where each plan $plan \in Plan$ specifies a sequence of actions believed by the agent to achieve the learning goals. Each p is the structural representation of knowledge points in the curious learning companion's memory M . For example, the partial plan structure in Figure 7.2 consists of four knowledge points: $k_1 : R(f : key \text{ is available}, a_i : take \text{ key}) = r^{add}$, $k_2 : R(f : key \text{ is available}, a_i : drive) = r^{pre}$, $k_3 : R(f : car \text{ at the garage}, a_i : drive) = r^{pre}$, and $k_4 : R(f : car \text{ at the garage}, a_i : drive) = r^{del}$. P is dynamically formed based on the current memory M . A newly discovered knowledge point k can be easily added into the curious learning companion's plan structure according to the relationship specified in k .

Figure 7.2: The partial plan structure for the goal of “driving to the supermarket”.

With the plan structured specified above, the curious learning companion can perform planning to select an applicable action given the learning goal and the current world state s_i [102]. This allows the agent to have a similar reasoning process with the human learner. Further more, in order to make the agent appraise world events and predict future environmental states at the same knowledge level with the human learner, the agent should be able to update the plan structure in accordance to the human learner’s learning progress. Next, we will introduce the learning function that updates the agent’s plan structure when the human learner learns.

7.2.2.2 Learning

Learning enables the agent to add in new knowledge in the memory M and update the plan structure $Plan$ when the human learner learns. evt_k triggers the learning function if the perceived knowledge point k is not in M . The curious learning companion update $Plan$ based on the relationship described by k : if a knowledge point gives information about the precondition of an action, i.e., $R(f, a) = r^{pre}$, then $pre(a) \leftarrow pre(a) \cup f$. If a knowledge point gives information about the effect of an action, i.e., $R(f, a) = r^{add}$, then $add(a) \leftarrow add(a) \cup f$; or $R(f, a) = r^{del}$, then $del(a) \leftarrow del(a) \cup f$.

With the internal functions discussed above, the agent is able to maintain a similar world model to the user’s one, which is the basis for detecting user’s knowledge gaps. In other words, the agent’s world model is an estimation of the user’s world model. Next, we will introduce the curious function which is at the core of knowledge gap detection.

7.2.3 The Curious Function: Stimulus Detection

According to Loewenstein [65], curiosity can be viewed from an information-gap perspective, which states that curiosity arises when a person’s attention becomes focused on a gap in his/her knowledge. From this viewpoint, Berlyne’s collative variables [8] can

be explained as reflecting multi-dimensional facets of a person’s knowledge gaps. For example, novelty refers to something unknown (missing information) and surprise arises when an expectation is violated (information mismatch). Hence, in order to detect user’s knowledge gaps, we focus on implementing one of the curious functions, i.e., stimulus detection (refer to Section 3.3), for the curious learning companion.

With the plan-based knowledge representation, three most salient collative variables in the VLE are captured, including novelty, surprise, and uncertainty. An assumption here is that since the agent maintains a world model that estimates the user’s world model, the agent’s knowledge gaps should also reflect the user’s knowledge gaps. Hence, the collative variables detected by the agents, when presented to the user, may elicit the user’s curiosity as well. Next, we will discuss the detection of the three collative variables in detail.

Novelty refers to something new. For instance, a child would be attracted to a toy that he/she has not seen before. The appraisal of novelty involves of a comparison between the current stimuli with the previously experienced ones. Intuitively, when the curious learning companion perceives a fact f , if $\forall k : R(f', a)$ in M satisfies $obj(f') \neq obj(f)$ and $obj(a) \neq obj(f)$, then the object $obj(f)$ is novel to the curious learning companion. In other words, an object o is novel if the curious learning companion has no priori knowledge (facts or actions) about it. A novel o indicates that the knowledge points associated with o do not exist in M and hence create a knowledge gap, which leads to the curious learning companion’s curiosity. Let $O^c \subseteq O$ denotes the set of objects that the curious learning companion is currently curious at. If object o is novel, then $o \in O^c$.

Surprise implies a violation of expectation. For instance, a child would be surprised if a toy car goes backward when the forward button is pressed. The appraisal of surprise involves comparisons between the expectation and the true facts. When the human learner performs action a at state s_i , the curious learning companion can make predictions about the next state, denoted by $F' = s_i \cup add(a) \setminus del(a)$. If the true facts in state s_{i+1} do not match with the predicted set F' , then $\forall f \in (s_{i+1} \setminus F') \cup (F' \setminus s_{i+1})$, the curious learning companion is surprised at $obj(f)$. This means that an object o is surprising if it does not follow the curious learning companion’s expectation when the human learner’s actions are performed upon o . Surprise indicates that the curious learning companion is

not clear about how an object is affected by an action, which also reflects a knowledge gap between the agent and the environment, leading to the curious learning companion's curiosity. If o is surprising to the agent, then $o \in O^c$.

Uncertainty arises when there is difficulty in selecting a response to a stimulus. For example, a child would feel uncertain about which road to choose for a toy car when insufficient information is provided for two roads. The appraisal of uncertainty involves comparisons among multiple competing responses. Let $A' = \{a | pre(a) \subseteq s_i\}$ denote a set of possible actions the curious learning companion believes that can be applied in the current state s_i , according to its current plan $plan$ for the learning goal $\mathcal{G}$. If $|A'| \geq 2$, then for $\forall a \in A'$, the curious learning companion is uncertain at $obj(a)$. This indicates that the curious learning companion feels uncertain when multiple actions are applicable in one state. We assume that actions have different preconditions. Otherwise, the human learner may easily get confused on how to act. Hence, uncertainty indicates that the precondition of certain actions is not complete and reveals a knowledge gap between M and K , which leads to the curious learning companion's curiosity. If o leads to uncertainty, then $o \in O^c$.

Table 7.1: The exemplary question templates for the curious learning companion.

Collative Variable	Question Templates
Novelty	Look! What is o over there? Let's go have a look! Hey! There is a o . What is it used for? Let's go find out!
Surprise	I expected f ($f \in F'$), but it happened to be f' ($f' \notin F'$). What can be the reason? It is really strange that f' ($f' \notin F'$) happened. Why? It surprises me! I don't know why a failed.
Uncertainty	I am not sure what will happen next. What do you expect to happen?

evt_s triggers the curious learning companion's curiosity appraisal process, i.e., the curious function. Once evt_s happens, the curious learning companion performs the curious function for discovering stimuli originated from novelty, surprise, and uncertainty. If the set of curiosity arousing objects is not empty, the curious learning companion will demonstrate curiosity by asking thought-provoking questions based on predefined question templates (Table 7.1), which brings the human learner's attention to the potential knowledge gaps and may elicit the human learner's curiosity.

The agent’s curiosity is quenched through the learning function when the user learns a new knowledge point. If a fact f is learnt by the curious learning companion, the agent removes the objects related to the fact and action described by k from the curious object set O^c .

The pseudo-code for the curious learning companion’s two concurrent processes: the curious function and the learning function is captured in Algorithm 4. It can be analyzed from Algorithm 4 that the major computational cost is at the curious function. Assuming that the maximum number of facts the agent can perceive at a time is m , the maximum number of knowledge points in the agent’s memory is n , and the number of agent’s action is z , then according to Algorithm 4 (Line 3-22), the maximum computational complexity of the curious function is $mn + m + z$.

7.3 Performance Evaluation of the Curious Learning Companion

In this section, we present the experimental details and the results obtained. This section starts with an illustration of the implemented system, and then give the hypotheses and field study setup. After that, the experimental results are discussed to analyze the impact of curiosity.

7.3.1 Implementation in Virtual Singapura

The proposed curious learning companion has been implemented and incorporated into Virtual Singapura (VS). An introduction of VS has been given in Section 6.3.1. The experiment is conducted in the underground scene of VS, where learners will shrink their avatars to ant size and jump into an ant hole to access the underground world (Figure 7.3). The goal for the human learners is to collect necessary materials (water and mineral salts) and figure out how to send them into the root of the plant in order to save Uncle Ben’s dying banana tree. Through this game, learners will learn the key concepts in plant transport systems, such as osmosis, diffusion, and active transport. The virtual world of VS and the virtual learning companion were implemented with Torque 3D engine.

Figure 7.4 presents an overview of the curious learning companion architecture. We can see that the learner acts in the virtual world and the agent listens to two types of

Figure 7.3: An illustration of the virtual companion’s curious questions: “What does osmosis transport? Let’s ask the Sage.”

events in the environment. evt_s may trigger two functions: (1) the curious function to discover curiosity stimuli and ask relevant questions, and (2) the planning function to suggest actions for the learner. evt_k occurs when a knowledge point is discovered, which triggers the learning function to update the agent’s plan structure and to satisfy curiosity.

In the VLE, learning concepts are visualized as virtual objects, the states of which are changed through actions. For example, the concept *water* is visualized as a water molecule object in the virtual world, and the location of water changes from ground to root if the action *osmosis* is applied on the water molecule object. Learning goals can be achieved by following a correct plan of actions. For example, to complete the goal *bring water into root*, the correct plan is a_1 : *collect water molecules*, a_2 : *collect partially permeable membrane*, a_3 : *adjust water ratio*, and a_4 : *osmosis*. The VLE provides knowledge points that can be learnt by human learners to form plans. For example, an important knowledge point that is required to form the plan for the task *bring water into root* is *partially permeable membrane is required for osmosis*. In a VLE, a knowledge point can be acquired from other knowledgeable non-player characters.

Figure 7.4: The architecture of the curious learning companion.

evt_s happens when the world state changes due to the learner's actions. This will trigger the agent's curiosity appraisal process to discover curiosity stimuli in the environment. When the agent's curiosity arises, it will demonstrate curious behaviors by asking thought provoking questions to the learner and encourage the learner to actively look for answers. For example, when surprise arises upon the failure of action *osmosis*, the companion will show curiosity and ask "It surprises me! I'm wondering why osmosis failed." Another example is shown in Figure 7.3.

If no curiosity stimuli is discovered, the agent will assume its current plan to be flawless and select the next applicable action. The agent shares its plan with the learner through dialogs. For example, when the agent select the next action *osmosis*, it will inform the learner saying that "Hey, my plan is to get into the osmosis hole now."

Once the learner acquires a new knowledge point, evt_k triggers the agent's learning process to add the new knowledge point into its knowledge base, which contributes to forming better plans. For example, when the Sage informs that osmosis happens when water ratio in ground is larger than that in the root. The agent will incorporate $f:water\ ratio\ ground > root$ into $Pre(a_4: osmosis)$.

7.3.2 Hypotheses

To our best knowledge, there are no available curious learning companions in the literature to make comparisons. Hence, we built two versions of learning companions to

evaluate the performance of a curious learning companion:

- *Basic learning companion (BLC)*: a reactive agent without the curious function and offers basic companionship by following the learner and watch out world events for the learner. For example, if an osmosis door is perceived, the basic learning companion says “Look! There is an osmosis door in front of us!”.
- *Curious learning companion (CLC)*: an agent with the curious function, which offers not only basic companionship but also learning and curiosity appraisal ability to direct learners’ attention to knowledge gaps.

We followed Kim’s methodology for evaluating pedagogical agents [57] and studied the impact of a CLC on four aspects: learning, agent persona, self-efficacy, and interest. Our general hypotheses are:

H_1 : A CLC can lead to better learning gains than a BLC.

H_2 : Learners will perceive the persona of a CLC more favorably than that of a BLC.

H_3 : Learners who play with a CLC will achieve higher self-efficacy than those who play with a BLC.

H_4 : Learners will feel more interested in learning the plant transport systems with a CLC than with a BLC.

7.3.3 Field Study

31 graduate students in a local university voluntarily participated in the study. The participants included 21 male and 10 female. The average age of the participants was 27.52 ($S=2.85$). The participants were randomly divided into an experiment group and a control group. In the experiment group, 16 participants played VS with the CLC. In the control group, 15 participants played VS with the BLC. In the study, each subject played with an experimenter (i.e., the experiment designer) rather than with a peer. We chose this setting to avoid possible confounding effects due to the participants’ different playing patterns and priori knowledge.

The study includes one session of 50 minutes, which is divided into three parts: pre-test (10 min), game-playing (30 min), and post-test (10 min). Before playing the game, the participants were given a pre-test (refer to Appendix A) on plant transport systems

to evaluate the participants' priori knowledge. The maximum total score of the pre-test was worth 9 points. After receiving a brief introduction to the game, the participants played VS till they finished all the tasks. Then, they took a post-test (refer to Appendix B), which was the same with the pre-test, to evaluate their learning gains. They also filled out a questionnaire with 23 questions (refer to Appendix C), which was designed to evaluate their subjective feeling of the agent persona, self-efficacy, and interest. The questionnaires were adapted from Kim's scales [57] to ensure the validity. Each item was scaled from 1 (Strongly disagree) to 5 (Strongly agree).

7.3.4 Experiment Results

This section presents the results obtained from the field study.

7.3.4.1 Learning Gains

Learning gains was objectively measured by the difference between the pre-test and post-test scores. F-test and One-tailed T-test were conducted with the two groups. Table 7.2 shows that the F-test result supports equal variance of the two groups ($p=0.82>0.05$) and the one-tailed T-test reveals a significant main effect of the two learning companions on learning gains ($p=0.0016<0.05$). Hence, the results support H_1 that the CLC can lead to better learning gains than the BLC.

7.3.4.2 Agent Persona

Agent persona was assessed with the Agent Persona Instrument (API) [6]. According to API, agent persona includes four sub-measures: facilitating learning, credible, human like, and engaging. Hence, H_2 is decomposed into four sub-hypotheses:

H_{2a} : A CLC can better facilitate learning than a BLC.

H_{2b} : A CLC is more credible than a BLC.

H_{2c} : A CLC is more human-like than a BLC.

H_{2d} : A CLC is more engaging than a BLC.

Table 7.3 presents the results for agent persona. In complementary to the objective measurement of learning by learning gains, facilitating learning measures learning subjectively. For facilitating learning, the F-test result supports equal variance of the

two groups ($p=0.87>0.05$). The one-tailed T-test reveals a significant main effect of the two learning companions on facilitating learning ($p=4.5*10^{-6}<0.05$). Hence, the results support H_{2a} that the CLC can better facilitate learning than the BLC.

From the perspective of credible, the F-test result supports equal variance of the two groups ($p=0.81>0.05$). The one-tailed T-test reveals a significant main effect of the two learning companions on their perceived credibility ($p=0.0005<0.05$). Hence, the results support H_{2b} that the CLC is perceived to be more credible than the BLC.

From the perspective of human-like, the F-test result supports equal variance of the two groups ($p=0.54>0.05$). The one-tailed T-test reveals that there is a significant main effect of the two learning companions on participants' perception of whether the learning companions are human-like ($p=0.03<0.05$). Hence, the results support H_{2c} that the CLC is perceived to be more human-like than the BLC.

From the perspective of engaging, the F-test result supports equal variance of the two groups ($p=0.20>0.05$). The one-tailed T-test reveals a significant main effect of the two learning companions on participants' perception of whether the agents are engaging ($p=0.001<0.05$). Hence, the results support H_{2d} that the CLC is more engaging than the BLC.

7.3.4.3 Self-efficacy

Self-efficacy is defined as individuals' belief in their competency of performing a particular task [3]. In this study, self-efficacy refers to the learners' belief about their competency in the knowledge of plant transport systems. The results for self-efficacy are presented in Table 7.4. The F-test result supports equal variance of the two groups ($p=0.09>0.05$). The one-tailed T-test reveals a significant main effect of the two learning companions on participants' self-efficacy ($p=0.02<0.05$). Hence, the results support H_3 that learners who play with the CLC can achieve higher self-efficacy than those who play with the BLC.

7.3.4.4 Interest

Interest is defined as a disposition organized through experience which impels an individual to seek out particular objects, activities, or goals for attention or acquisition [27].

In this study, interest refers to learners' disposition towards learning the topic of plant transport systems with the learning companions. The results for interest are presented in Table 7.5. The F-test result supports equal variance of the two groups ($p=0.38>0.05$). The one-tailed T-test reveals that there is a significant main effect of the two learning companions on participants' interest ($p=0.005<0.05$). Hence, the results support H_4 that learners feel more interested in learning the plant transport systems with the CLC than with the BLC.

7.3.5 Discussion

The mean scores of the control group and experiment group for the seven aspects are shown in Figure 7.5. The results confirm that the CLC can help learners achieve more learning gains (H_1), better facilitate learning (H_{2a}), be more credible (H_{2b}), be more human-like (H_{2c}), be more engaging (H_{2d}), better improve learner's self-efficacy (H_3), and make learners feel more interested (H_4) than the BLC. The CLC achieves better results probability because the agent is driven by curiosity to learn, which follows a similar learning process as the learner. Hence, the CLC provides hints that are aware of both the environmental context and learners' learning progress. Whereas, the BLC does not actually "learn" with the learners, which can only provide hints that are environmental context aware but not learning progress aware.

Another possible reason is that a CLC may stimulate learners' curiosity through questions it poses. By presenting the gaps of knowledge with thought-provoking questions, the learners can become motivated to find answers and actively inquire into the subject in order to resolve their curiosity. In this way, inquiry-based learning [20] is established and learners learn in a question-driven and open-ended process.

From our observations, the CLC helps more with learners who are less active in asking themselves questions and looking for answers. In this case, the CLC acts more like a leader and learners follow the leader to complete tasks. In the case with learners who take more initiative in questioning and answering, learners take the role as leaders and the CLC acts like a follower who looks out for information probably missed by learners.

Figure 7.5: The mean scores of control and experiment group for seven aspects: learning gains, facilitating learning, credible, human-like, engaging, self-efficacy, and interest.

7.4 Summary

In this chapter, we presented a curious learning companion. It is inspired by Loewenstein’s information-gap theory on curiosity, which states that curiosity can be viewed as arising when attention becomes focused on a gap in one’s knowledge. Hence, introducing curiosity into virtual learning companions may allow the agents to discover users’ knowledge gaps and direct their attention towards those knowledge gaps. The curious learning companion is proposed based on the generic computational model of curiosity. The computational model of curiosity is implemented based on plan-based representations that is at the core of a number of complex real-time autonomous systems. With plan-based representations, the agent is able to appraise three collative variables that are most salient in the VLE, including novelty, surprise, and uncertainty. The curious learning companion is coded in Virtual Singapura and carefully studied with human users. The evaluation results show that the virtual learning companion’s curiosity significantly improves human learners’ learning experience from several aspects, including learning gains, self-efficacy, and interest. Also, the results support that human learners perceive a curious learning companion’s persona more favorably.

Till now, we have finished Part III of this thesis, in which the proposed computational framework of curiosity is studied in the domain of virtual worlds for educational purposes.

The experimental results show that curiosity brings positive effects on the performances of different pedagogical agents. In the next part of this thesis, we will summarize the major contributions and suggest future research directions.

Algorithm 4: PSEUDOCODE FOR THE CURIOUS LEARNING COMPANION

```

1   $O^c \leftarrow \emptyset$  // Initialize curious object set;
2  while true do // Run thread for curiosity appraisal
3      if  $evt_s$  occurs then // Listen to  $evt_s$ 
4          Perceive current state  $s_i$ , retrieve the human learner's action  $a$  at  $s_{i-1}$ ;
5          foreach  $f \in s_i$  do // Check novelty
6              foreach  $R(f', a) \in B$  do
7                  if  $obj(f) \neq obj(f') \& obj(f) \neq obj(a)$  then
8                       $O^c \leftarrow O^c \cup obj(f)$ ;
9                  end
10             end
11         end
12         Retrieve prediction  $F'$  at  $a$  // Check surprise;
13         if  $s_i \neq F'$  then
14             foreach  $f \in (s_i \setminus F') \cup (F' \setminus s_i)$  do
15                  $O^c \leftarrow O^c \cup obj(f)$ ;
16             end
17              $O^c \leftarrow O^c \cup obj(a)$ ;
18         end
19          $A' \leftarrow \emptyset$  // Initialize applicable action set;
20         foreach  $a \in \text{current } p$  do // Check uncertainty
21             if  $pre(a) \subseteq s_i$  then  $A' \leftarrow A' \cup a$ ;
22             ;
23         end
24         if number of elements in  $A' \geq 2$  then
25             foreach  $a \in A'$  do  $O^c \leftarrow O^c \cup obj(a)$  ;
26             ;
27         end
28         if  $O^c \neq \emptyset$  then // Demonstrate curiosity
29             Generate questions on  $O^c$ ;
30         end
31         else // Plan next action for the human learner
32             Plan next action  $a'$  based on  $Plan$ ;
33         end
34     end
35 end
36 while true do // Run thread for learning
37     if  $evt_k$  occurs then // listen to  $evt_k$ 
38         Perceive the knowledge point  $k : R(f, a)$  ;
39         if  $R(f, a) \notin B$  then // Update plan structure
40             if  $R(f, a) = r^{pre}$  then  $pre(a) \leftarrow pre(a) \cup f$ ;
41             ;
42             else if  $R(f, a) = r^{add}$  then  $add(a) \leftarrow add(a) \cup f$ ;
43             ;
44             else if  $R(f, a) = r^{del}$  then  $del(a) \leftarrow del(a) \cup f$ ;
45             ;
46             // Remove from the curious object set;
47             if  $obj(f) \in O^c$  then  $O^c \leftarrow O^c \setminus obj(f)$ ;
48             ;
49             if  $obj(a) \in O^c$  then  $O^c \leftarrow O^c \setminus obj(a)$ ;
50             ;
51         end
52     end
53 end
    
```

Table 7.2: The performance comparison between the basic learning companion and the curious learning companion on learning gains

Group	Control	Experiment
Mean	2.4	5.25
STD	2.56	2.41
F	p=0.82	
T	p=0.0016	

Table 7.3: The performance comparison between the basic learning companion and the curious learning companion on agent persona

	Facilitating Learning		Credible	
Group	Control	Experiment	Control	Experiment
Mean	2.16	3.58	2.52	3.50
STD	0.72	0.76	0.76	0.72
F	p=0.87		p=0.81	
T	p=4.5*10 ⁻⁶		p=0.0005	
	Human-like		Engaging	
Group	Control	Experiment	Control	Experiment
Mean	2.85	3.44	2.47	3.53
STD	0.92	0.78	1.01	0.72
F	p=0.54		p=0.20	
T	p=0.03		p=0.001	

Table 7.4: The performance comparison between the basic learning companion and the curious learning companion on self-efficacy

Group	Control	Experiment
Mean	3.60	4.14
STD	0.87	0.55
F	p=0.09	
T	p=0.02	

Table 7.5: The performance comparison between the basic learning companion and the curious learning companion on interest

Group	Control	Experiment
Mean	2.63	3.21
STD	0.52	0.66
F	p=0.38	
T	p=0.005	

Part IV

Conclusion

Chapter 8

Conclusion and Future Directions

8.1 Thesis Summary

This thesis has made two major contributions in computational curiosity and its applications, including a computational framework of curiosity and its novel application in the domain of virtual worlds for education. The computational framework of curiosity consists of a generic model of curiosity, a curiosity-driven learning algorithm, and a curiosity-driven recommendation algorithm. The application in virtual worlds based education includes the studies on a curious peer learner and a curious learning companion. In this section, we will summarize each of the contributions in detail.

8.1.1 Computational Framework of curiosity

In the computational framework of curiosity, there are three sub-contributions: a computational model of curiosity, a curiosity-driven learning algorithm, and a curiosity-driven recommendation algorithm.

8.1.1.1 Computational Model of Curiosity

In this thesis, a generic computational model of curiosity is developed for intelligent agents. This computational model is built based on Berlyne's and Wundt's theories of curiosity in psychology. Berlyne theorized curiosity as a process of stimulus selection that can be governed by a set of collative variables, such as novelty, surprise, uncertain, conflict, and complexity. Wundt postulated an inverted U-shape relationship between the level of curiosity stimulation and the hedonic value, and pointed out two emotions

that are closely related to curiosity, i.e., boredom and anxiety. The computational model of curiosity transposes these theories into a set of abstract functions and the interactions between them. Two curious functions are proposed, i.e., stimulus selection and interest evaluation, which serve as meta-level decision making functions that regulate other internal functions of the agent, such as learning and decision making. Modeling agent models with abstract functions makes them general enough to allow different implementations in diverse application contexts.

8.1.1.2 Curiosity-driven Extreme Learning Machine

A curiosity-driven learning algorithm is developed based on the generic computational model of curiosity. The computational model of curiosity is implemented in a fast neural learning algorithm: extreme learning machine classifier. The curious extreme learning machine classifier (C-ELM) treats each input data as a curious stimulus and performs curiosity appraisal towards each input data using four collative variables: novelty, uncertainty, conflict, and surprise. Based on the curiosity appraisal results, three learning strategies can be chosen from to deal with the input data, including neuron addition, neuron deletion, and parameter update. C-ELM enhances the traditional ELM algorithm with three major contributions: (1) the evolving capability, which determines optimal network structure dynamically based on training data, (2) reduced random effect, which is achieved by selecting RBF centers based on data instead of random assignment, (3) a novel neuron deletion strategy, which is based on conflict resolution. Empirical study of C-ELM shows that the proposed approach leads to compact network structures and generates better generalization performance with fast response, comparing with traditional ELM and other popular classifiers.

8.1.1.3 Curiosity-driven Recommendation

A curiosity-driven recommendation algorithm is developed based on the generic computational model of curiosity. The idea of introducing curiosity into recommender systems is inspired by the common observed phenomenon in real life that we are often interested by our friends' abnormal behaviors. This is generally known as social curiosity in psychology. Hence, recommending items that have been given an unexpected rating by a

friend may cause the target user's interest. For the curious recommender agent, the computational model of curiosity is implemented based on the popular predictive model by matrix factorization, due to its high recommendation accuracy and efficiency in dealing with large scale user-item rating matrices. The curious recommender agent is carefully studied with three data sets (Movie, Music, and Book) crawled from <http://Douban.com>. The experimental results showed that the curious recommender agent successfully recommends surprising items and significantly enhances the diversity of recommendation lists.

8.1.2 Application in Virtual Worlds for Education

In this thesis, the computational framework of curiosity is studied in the application domain of virtual worlds for educational purposes. Curiosity is realized in two types of pedagogical agents: a curious peer learner and a curious learning companion.

8.1.2.1 Curious Peer Learner

A curious peer learner is realized based on the proposed computational model of curiosity. Virtual peer learners are non-player characters to provide a believable virtual learning environment for users. Curiosity is the motivation that drives human learning behaviors. Hence, modeling curiosity for virtual peer learners may allow them to provide believable and human-like behaviors in a learning context. The generic computational model of curiosity is implemented using the computational theories of concept maps and reinforcement learning. Concept maps allow the knowledge representations to be easily interpreted by the human learners and the virtual agents as well. Both of the two curious functions, i.e., stimulus selection and interest evaluation, are realized in the curious peer learner. Three collative variables are employed for stimulus selection, including novelty, conflict, and complexity. The emotion states derived from interest evaluation generate intrinsic rewards to update the agent's Q-table and regulate its learning strategies. Experiment studies through computer simulations show positive impact of curiosity on the performance of virtual peer learners. Simulation results demonstrate that curiosity-related emotions can guide the virtual peer learner to behave as naturally

as a real student does. Also, the comparison between a curious peer learner and a non-curious peer learner show that the curious peer learner can demonstrate higher tendency for exploration (exploration breadth), as well as higher average knowledge acquisition efficiency (exploration depth).

8.1.2.2 Curious Learning Companion

A curious learning companion is realized based on the proposed computational model of curiosity. Virtual learning companions are pedagogical agents that take a peer metaphor to simulate peer interactions with the users. Modeling curiosity for virtual learning companions may allow them to detect knowledge gaps and ask relevant questions to stimulate users' motivation to learn. The computational model of curiosity is implemented using plan-based representations that is at the core of a number of complex real-time autonomous systems. With plan-based representations, the agent is able to appraise three collative variables that are most salient in the virtual learning environment, including novelty, surprise, and uncertainty. Based on the collative variables, the curious learning companion can ask relevant questions to users so as to bring their attention towards potential knowledge gaps and motivate them to actively explore the virtual learning environment. The curious learning companion is coded in Virtual Singapura and carefully studied with human users. The evaluation results show that the virtual learning companion's curiosity significantly improves human learners' learning experience from several aspects, including learning gains, self-efficacy, and interest. Also, the results support that human learners perceive a curious learning companion's persona more favorably.

8.2 Future Directions

The studies conducted in this thesis could be extended in various aspects. A few of these possible research directions are discussed in this section from two viewpoints: the modeling aspects of computational curiosity and the other research areas that computational curiosity will impact.

8.2.1 Modeling Aspects of Computational Curiosity

In this subsection, we will discuss several interesting aspects that can be considered in the modeling of computational curiosity.

8.2.1.1 Interactions between Collative Variables

In the proposed computational model of curiosity, the collative variables are treated as independent factors that stimulate curiosity. This model neglects the inter-influences between the collative variables which have been discussed by Berlyne [8] from the field of psychology. Berlyne postulated that novel stimuli can lead to conflict when a new pattern of stimulus is sufficiently similar to several familiar stimulus patterns, which may induce many incompatible responses. Also, novelty has a connection with complexity. For example, a stimulus with short-term novelty will have a higher degree of temporal complexity than purely repetitive patterns. Alternatively, a higher degree of synchronous complexity may induce a higher degree of relative novelty. Moreover, complexity is also associated with uncertainty and conflict. For example, complex patterns with more parts can assume a larger amount of alternative forms and the degree of uncertainty can be affected by the number of classes in these alternative forms; each subelement of a complex figure may induce different responses that are incompatible with each other, leading to a higher level of conflict.

Modeling the interactions between the collative variables in the stimulus selection function may yield a more accurate curiosity model for the intelligent agents. However, the interactions between the collative variables are rather vague and complicated. Hence, it is difficult to define the interaction weights using expert knowledge. A possible solution is to learn the interaction weights between the collative variables from data through a machine learning approach.

8.2.1.2 Neuroscience Inspired Modeling Approaches

The current computational model of curiosity is mainly inspired by the drive theory and optimal arousal theory from the psychological viewpoint. Recently, researchers argued that the psychological models failed to reconcile whether the reward of obtaining new knowledge is mediated through a feeling of “deprivation”, as defined by Loewenstein [65],

Table 8.1: An illustration of the interest deprivation (I/D) theory of curiosity

		Wanting	
		Low level	High level
Liking	High level	Curiosity as a feeling of “interest” (Aesthetic appreciation)	Curiosity as a feeling of “deprivation” (Perceptual/conceptual/fluency)
	Low level	Ambivalent disinterest or boredom (Spontaneous alternation or novelty seeking)	Need for uncertainty clarification (Need for cognitive closure; morbid or lurid curiosity)

or a feeling of “interest”, as defined by Spielberger and Starr [128]. Hence, researchers began to integrate the neurological aspects of reward, wanting, and liking into a new way to understand curiosity, one that is explained by biological processes. Litman [63] developed interest deprivation (I/D) theory of curiosity that incorporates the neuroscience of “wanting” and “liking”, which are two systems hypothesized to underlie motivation and affective experience for a broad class of appetites. Litman theory can be summarized in table 8.1. It can be seen that according to this theory, the different states of curiosity are determined by the corresponding levels of the two factors: wanting and liking. This theory motivates the modeling of computational curiosity from a neuroscience inspired approach, which may be considered in the future works to provide a more comprehensive computational model of curiosity.

8.2.1.3 Recognition and Response to Users’ Curiosity

The current research on computational curiosity mainly focused on designing models for intelligent agents to simulate human curiosity. However, these models may not work well in the context of human-agent interaction if they omit users’ current state of curiosity. For example, the curious learning companion, which detects a user’s knowledge gaps and poses these knowledge gaps to the user through questions, may be redundant if the user has noticed the knowledge gaps himself/herself and become curious already. Hence, one interesting research direction for computational curiosity which has not been explored is to model user’s curiosity. This topic is especially important for designing next-generation human-agent collectives in areas such as embodied conversational agents, computer-based education, and artificial companions. The ability of intelligent agents to recognize and

respond to users' curiosity provides opportunities to enhance user experiences in human-computer interactions by avoiding boring and anxiety-inducing topics, or by stimulating users' curiosity appropriately. This research may also enrich the current understanding of human curiosity in psychology.

In order to detect user's curiosity states unobtrusively, many methods could be adopted. Firstly, physiological information such as EGG, heart rate, etc. can be detected by wearable equipments [27, 28]. Secondly, facial expressions can be captured by mounting a video camera and pattern recognition methods can be applied to detect different affect states associated with the facial expressions [9]. Thirdly, the user's behavior patterns in the game environment can also be a good indicator of the user's affect states [73]. For example, over frequent mouse clicking can be an indicator of anxiety and a random clicking of irrelevant objects can be an indicator of boredom.

8.2.1.4 Curiosity-based Decision Making

One interesting issue with computational curiosity is the risk management in curiosity-based decision making. As curiosity often leads to explorations, sometimes the agent or human being may be exposed to the possibility of being harmed or causing undesirable consequences to others. Hence, curious machines should operate under the protection of proper risk management systems so that they will not harm themselves. Ethical boundaries should also be defined for curious agents so that they will not intrude on the privacy of the users or other agents.

8.2.2 Research Areas that Computational Curiosity will Impact

From the literature review, we have summarized four major application domains that existing research on computational curiosity has impacted, including machine learning, robotics, artificial creativity, and games. In this thesis, we mainly focused on three areas of application: machine learning, recommender system, and virtual worlds for education. However, as a novel characteristic for intelligent agents, the impact of curiosity is not limited to these research areas. Next, we will outline other research areas that computational curiosity may impact.

8.2.2.1 Affective Computing

Affective computing is computing that relates to, arises from, or deliberately influences emotion or other affective phenomena [92]. It requires multidisciplinary knowledge such as psychology, cognitive science, computer science, and engineering. Affective computing is gaining rapid popularity and has great potential in the next generation of human-computer interfaces. Curiosity is closely related to emotional constructs such as “fear”, “pleasure”, “boredom”, and “anxiety” [65]. Computational curiosity offers a new dimension from which emotions can be appraised, apart from the consequences of events, the actions of agent, and the characteristics of objects [85]. The consideration of computational curiosity in affective modeling is especially interesting in learning contexts and social contexts, where curiosity-related emotions significantly influences the believability and performance of emotional agents.

8.2.2.2 Artificial Companions

Artificial companions, designed to develop a close and long-term human computer relationship, have emerged in the latter half of the 2000s. Two key words, *close* and *long-term*, have been guiding the development in this field. Researchers are working on the design of believable human-computer interfaces to provide close interactions (e.g. embodied conversational agent) and robust memory architectures to sustain long-term relationships [9, 140]. Computational curiosity can be an important dimension to be studied in artificial companions for enhancing both the closeness of interactions and the possibility for long-term relationships. The potential for computational curiosity in creating a closer human-computer relationship draws evidence from psychological findings that curiosity plays an important role in promoting the intimacy of interpersonal relationships in social context. A curious artificial companion can be more responsive; may infuse more novel twists of excitement into interactions, and might induce a natural flow of engagement between the interaction discourses. As for promoting long-term relationships, curiosity can be a motivational force to learn more about the partner and develop a historical knowledge base through interactions. A curious artificial companion may be more interested to know the partner; may be more inquisitive to novel changes of the partner; and may incorporate information of the partner into part of the cognitive development of the companion itself.

8.2.2.3 Persuasive Technology

Persuasive technology deals with the use of computing systems, devices or applications to gradually change a person's attitudes or behavior [25]. This technology has the potential to bring constructive changes in health science, safety and education. Examples include a digital doll to persuade kids to eat fruit and vegetables, and a virtual coach to persuade the elderly to exercise more [24]. Understanding users' curiosity and properly infusing curiosity stimuli into the human-computer interaction process can potentially help intelligent agents achieve persuasive goals. For example, if a sales agent can successfully elicit the customer's curiosity in a product, there will be a higher chance for this product to be sold. Curiosity has been harnessed to "persuade" programmers to increase the correctness in end-user programming [141].

8.2.2.4 Agent Negotiation

Negotiation is a process that involves two or more parties to reach an agreement. This mechanism has received increasing attention in multi-agent systems for managing inter-agent dependencies in real time [44]. Traditional implementations of negotiation process focused on its rational aspects to build consensus. Recently, [14] argued that negotiation is a multifaceted process in which affect plays an important role. Computational curiosity has the potential to influence the human-agent negotiation process by promoting positive emotional states, responsiveness and engagement. Enabling a negotiation agent to understand the curiosity exhibited by a user may allow it to notice the unusual, surprising, or conflicting information offered by the user and reach agreements that are more socially optimal. A negotiation agent that can adapt its decision-making based on the users' curiosity may improve its chance of gaining more utility out of the final agreement.

8.2.2.5 Trustworthy Computing

Another important issue in multi-agent systems is trust management. It is useful in open and dynamic systems such as peer-to-peer systems, semantic Web, ad hoc networks, and e-commerce, etc. [96, 153]. Similar to negotiation, trust management is also closely related to emotion states [19, 122]. The motivational role of curiosity in building interpersonal relationships can contribute to the trust building between strangers

[50]. Computational curiosity can potentially enhance an agent’s judgement by making the agent more sensitive to novel, surprising, conflicting, and uncertain information presented in the environment.

Author's Publications

Journals

1. Qiong Wu and Chunyan Miao. Curiosity: From psychology to computation. *ACM Computing Surveys*, 46(2):18, 2013.
2. Qiong Wu and Chunyan Miao. Modeling curiosity-related emotions for virtual peer learners. *Computational Intelligence Magazine*, 8(2):50-62, 2013.

Conferences

1. Qiong Wu and Chunyan Miao. C-ELM: A curious extreme learning machine for classification problems. *The International Conference on Extreme Learning Machines (ELM2015)*, 2015.
2. Qiong Wu, Chunyan Miao, and Bo An. Modeling curiosity for virtual learning companions. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2014.
3. Qiong Wu, Zhiqi Shen and Chunyan Miao. Stimulating students' curiosity in virtual learning environments. *World Conference on Educational Media and Technology (EdMedia)*, 2013.
4. Qiong Wu, Xiaogang Han, Han Yu, Zhiqi Shen, Chunyan Miao. The innovative application of learning companions in virtual singapura. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.
5. Qiong Wu, Chunyan Miao, and Zhiqi Shen. A curious learning companion in virtual learning environment. *IEEE World Congress on Computational Intelligence (WCCI)*, 2012 (Best Paper Nomination).

6. Qiong Wu, Chunyan Miao, Xuehong Tao, and Martin Helander. A curious companion for elderly gamers. *Southeast Asian Network of Ergonomics Societies Conference (SEANES)*, 2012.

Bibliography

- [1] A. Al Mahmud, P. Dadlani, O. Mubin, S. Shahid, C. Midden, and O. Moran. iparrot: towards designing a persuasive agent for energy conservation. *Persuasive Technology*, pages 64–67, 2007.
- [2] G. Baldassarre. What are intrinsic motivations? A biological perspective. *In: Proc. of IEEE Conference on Development and Learning*, pages 1–8, 2011.
- [3] A. Bandura. *Self-efficacy: The exercise of control*. Ner York: W.H. Freeman, 1997.
- [4] A.G. Barto, S. Singh, and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. *In: Proc. of International Conference on Development Learn*, pages 112–119, 2004.
- [5] A.G. Barto, S. Singh, and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. *In: Proc. of International Conference on Development Learn*, pages 112–119, 2004.
- [6] A. Baylor and J. Ryu. The API (Agent Persona Instrument) for assessing pedagogical agent persona. *In Proc. of The World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 448–451, 2003.
- [7] Daniel E Berlyne. *Conflict, arousal, and curiosity*. McGraw-Hill Book Company, 1960.
- [8] D.E. Berlyne. *Conflict, arousal, and curiosity*. McGraw-Hill New York, 1960.
- [9] T.W. Bickmore and R.W. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Transaction on Computer-Human Interaction*, 12(2):293–327, 2005.

- [10] G. Biswas, K. Leelawong, D. Schwartz, and N. Vye. Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, 19(3-4):363–392, 2005.
- [11] M. A. Boden. Computer models of creativity. *AI Magazine*, 30(3):23–34, 2009.
- [12] D. Boud, R. Cohen, and J. Sampson. *Peer Learning in Higher Education: Learning from & with Each Other*. ERIC, 2001.
- [13] K. Bradley and B. Smyth. Improving recommendation diversity. In *Proc. of the National Conference in Artificial Intelligence and Cognitive Science*, pages 75–84, 2001.
- [14] J. Broekens, C.M. Jonker, and J.J.C. Meyer. Affective negotiation support systems. *Journal of Ambient Intelligence and Smart Environments*, 2(2):121–144, 2010.
- [15] J.E. Brophy and T.L. Good. *Teacher-student relationships: Causes and consequences*. Rinehart & Winston, 1974.
- [16] W. Burgard, M. Moors, and F. Schneider. Collaborative exploration of unknown environments with teams of mobile robots. *Advances in plan-based control of robotic agents*, 2466:187–215, 2002.
- [17] D.A. Cohn, Z. Ghahramani, and M.I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.
- [18] C. Dede. Immersive interfaces for engagement and learning. *Science*, 323:66–69, 2009.
- [19] J.R. Dunn and M.E. Schweitzer. Feeling and believing: The influence of emotion on trust. *Journal of personality and social psychology*, 88(5):736–748, 2005.
- [20] D.C. Edelson, D.N. Gordin, and R.D. Pea. Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the Learning Sciences*, 8(3-4):391–450, 1999.

- [21] M.J. Eisen. Peer-based learning: A new-old alternative to professional development. *Adult Learning*, 12(1):9–10, 2001.
- [22] V. Fedorov. *Theory of optimal experiments*. Academic Press, 1972.
- [23] G. Feng, G.B. Huang, Q. Lin, and R. Gay. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Transactions on Neural Networks*, 20(8):1352–1357, 2009.
- [24] B.J. Fogg. Persuasive technologie. *Communications of the ACM*, 42(5):26–29, 1999.
- [25] B.J. Fogg. Persuasive technology: Using computers to change what we think and do. *Ubiquity*, 2002(5):27–29, 2002.
- [26] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proc. of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, 1996.
- [27] J.W. Getzels. The problem of interests: A reconsideration. *Supplementary Education Monographs*, 66:97–106, 1966.
- [28] J. Gratch. Emile: Marshalling passions in training and education. In *Proc. of International Conference on Autonomous Agents*, pages 325–332, 2000.
- [29] V. Graziano, T. Glasmachers, T. Schaul, L. Pape, G. Cuccu, J. Leitner, and J. Schmidhuber. Artificial curiosity for autonomous space exploration. *Acta Futura*, (4):41–51, 2011.
- [30] A.L. Harris and A. Rea. Web 2.0 and virtual world technologies: A growing impact on IS education. *Journal of Information Systems Education*, 20(2):137–144, 2009.
- [31] D.O. Hebb. *The organization of behavior*. New York: Wiley, 1949.
- [32] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

- [33] H. Hu, E. Real, K. Takamiya, M.G. Kang, J. Ledoux, R.L. Huganir, and R. Malinow. Emotion enhances learning via norepinephrine regulation of ampa-receptor trafficking. *Cell*, 131(1):160–173, 2007.
- [34] G.B. Huang and L. Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16):3056–3062, 2007.
- [35] G.B. Huang and L. Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16):3460–3468, 2008.
- [36] G.B. Huang, D.H. Wang, and Y. Lan. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122, 2011.
- [37] G.B. Huang, Q.Y. Zhu, K.Z. Mao, C.K. Siew, P. Saratchandran, and N. Sundararajan. Can threshold networks be trained directly? *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(3):187–191, 2006.
- [38] G.B. Huang, Q.Y. Zhu, and C.K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.
- [39] G.B. Huang, Q.Y. Zhu, and C.K. Siew. Real-time learning capability of neural networks. *IEEE Transactions on Neural Networks*, 17(4):863–878, 2006.
- [40] M.J. Jacobson, B. Kim, C. Miao, and M. Chavez. *Design for learning environments of the future*, chapter Design perspectives for learning in virtual worlds, pages 111–141. Springer US, 2010.
- [41] Mohsen Jamali and Martin Ester. A transitivity aware matrix factorization model for recommendation in social networks. *In Proc. IJCAI’11*, pages 2644–2649, 2011.
- [42] W. James. *Principles of psychology*. New York: Holt, 1950.
- [43] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, M.J. Wooldridge, and C. Sierra. Automated negotiation: prospects, methods and challenges automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

- [44] N.R. Jennings and M.J. Wooldridge. *Agent technology: foundations, applications, and markets*. Springer, 2002.
- [45] W.L. Johnson and J. Rickel. Steve: An animated pedagogical agent for procedural training in virtual environments. *ACM SIGART Bulletin*, 8(1-4):16–21, 1997.
- [46] W.L. Johnson, J.W. Rickel, and J.C. Lester. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial intelligence in education*, 11(1):47–78, 2000.
- [47] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [48] C. Karaoguz, E. Drix, D. anad Potapova, and M. Huelse. Curiosity driven exploration of sensory-motor mappings. *Deliverable for the IM-CLeVeR Spring School at the Capo Caccia Cognitive Neuromorphic Engineering Workshop*, pages 1–7, 2011.
- [49] T.B. Kashdan. *Curious?: Discover the missing ingredient to a fulfilling life*. Harper Paperbacks, 2010.
- [50] T.B. Kashdan, P.E. McKnight, F.D. Fincham, and P. Rose. When curiosity breeds intimacy taking advantage of intimacy opportunities and transforming boring conversations. *Journal of Personality*, 79(6):1369–1402, 2011.
- [51] T.B. Kashdan, P. Rose, and F.D. Fincham. Curiosity and exploration: Facilitating positive subjective experiences and personal growth opportunities. *Journal of Personality Assessment*, 82(3):291–305, 2004.
- [52] T.B. Kashdan and M.F. Steger. Curiosity and pathways to well-being and meaning in life: Traits, states, and everyday behaviors. *Motivation and Emotion*, 31(3):159–173, 2007.
- [53] David Kempe, Jon Kleinberg, èŞŁa Tardos, and ACM. Maximizing the spread of influence through a social network. *In Proc. of ACM SIGKDD’03*, pages 137–146, 2003.

- [54] S. Kennedy-Clark. Designing failure to encourage success: Productive failure in a multi-user virtual environment to solve complex problems. *In: Proc. of European Conference on Technology Enhanced Learning*, pages 609–614, 2009.
- [55] S. Kennedy-Clark. Pre-service teachers’ perspectives on using scenario-based virtual worlds in science education. *Computers & Education*, 57(4):2224–2235, 2011.
- [56] S. Kennedy-Clark and K. Thompson. What do students learn when collaboratively using a computer game in the study of historical disease epidemics, and why? *Games and Culture*, 6(6):513–537, 2011.
- [57] Y. Kim. *Pedagogical agents as learning companions: The effects of agent affect and gender on learning, interest, self-efficacy, and agent persona*. PhD thesis, The Florida State University, 2004.
- [58] R. Koster. *A theory of fun for game design*. Paraglyph Press, 2005.
- [59] B. Li and M.O. Riedl. An offline planning approach to game plotline adaptation. *In: Proc. of the 6th Conference on Artificial Intelligence for Interactive Digital Entertainment*, pages 45–50, 2010.
- [60] M. Li and P.M.B. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag New York Inc, 2008.
- [61] N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, 17(6):1411–1423, 2006.
- [62] N.Y. Liang, P. Saratchandran, G.B. Huang, and N. Sundararajan. Classification of mental tasks from EEG signals using extreme learning machine. *International Journal of Neural Systems*, 16(01):29–38, 2006.
- [63] J. Litman. Curiosity and the pleasures of learning: Wanting and liking new information. *Cognition & emotion*, 19(6):793–814, 2005.
- [64] J.A. Litman and C.D. Spielberger. Measuring epistemic curiosity and its diversive and specific components. *Journal of Personality Assessment*, 80(1):75–86, 2003.

- [65] G. Loewenstein. The psychology of curiosity: A review and reinterpretation. *Psychological Bulletin*, 116(1):75–98, 1994.
- [66] M.A. Lundeberg and S.D. Moch. Influence of social interaction on cognition: Connected learning in science. *Journal of Higher Education*, 66(3):312–35, 1995.
- [67] H. Ma, D. Zhou, C. Liu, M.R. Lyu, and I. King. Recommender systems with social regularization. In *Proc. of ACM international conference on Web search and data mining*, pages 287–296, 2011.
- [68] L. Macedo. Selecting information based on artificial forms of selective attention. In: *Proc. of European Conference on Artificial Intelligence*, pages 1053–1054, 2010.
- [69] L. Macedo and A. Cardoso. Towards artificial forms of surprise and curiosity. In: *Proc. of the European Conference on Cognitive Science*, pages 139–144, 1999.
- [70] L. Macedo and A. Cardoso. Modeling forms of surprise in an artificial agent. In: *Proc. of the Annual Conference of the Cognitive Science Society*, pages 588–593, 2001.
- [71] L. Macedo and A. Cardoso. Assessing creativity: the importance of unexpected novelty. In: *Proc. of the ECAI Workshop on Creative Systems: Approaches to Creativity in AI and Cognitive Science*, pages 31–37, 2002.
- [72] L. Macedo and A. Cardoso. Exploration of unknown environments with motivational agents. In: *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 328–335, 2004.
- [73] L. Macedo and A. Cardoso. The role of surprise, curiosity and hunger on exploration of unknown environments populated with entities. In: *Proc. of Portuguese Conference on Artificial Intelligence*, pages 47–53, 2005.
- [74] M.L. Maher, K. Merrick, and R. Saunders. Achieving creative behaviour using curious learning agents. In: *Proc. of AAAI Spring Symposium on Creative Intelligent Systems*, pages 26–28, 2008.

- [75] K. Merrick. Designing toys that come alive: curious robots for creative play. *In: Proc. of International Conference on Entertainment Computing*, pages 149–154, 2008.
- [76] K. Merrick. Modeling motivation for adaptive nonplayer characters in dynamic computer game worlds. *Computers in Entertainment*, 5(4):5. 1–32, 2008.
- [77] K. Merrick and E. Huntington. Attention focus in curious, reconfigurable robots. *In: Proc. of Australian Conference on Robotics and Automation*, 2008.
- [78] K. Merrick and M.L. Maher. *Motivated reinforcement learning: curious characters for multiuser games*. Springer, 2009.
- [79] K. Merrick, M.L. Maher, and R. Saunders. Achieving adaptable behaviour in intelligent rooms using curious supervised learning agents. *In: Proc. of Conference on Computer Aided Architectural Design Research in Asia*, pages 185–192, 2008.
- [80] Y. Miche, A. Sorjamaa, and A. Lendasse. OP-ELM: theory, experiments and a toolbox. *Lecture Notes in Computer Science*, pages 145–154, 2008.
- [81] S.J. Moorehead, R. Simmons, and W.L. Whittaker. Autonomous exploration using multiple sources of information. *In: Proc. of IEEE International Conference on Robotics and Automation*, pages 3098–3103, 2001.
- [82] H. Ngo, M. Luciw, A. Forster, and J. Schmidhuber. Learning skills from play: Artificial curiosity on a katana robot arm. *In: Proc. of International Conference on Neural Networks*, pages 1–8, 2012.
- [83] J.D. Novak and D.B. Gowin. *Learning how to learn*. Cambridge University Press, 1984.
- [84] M. Ogino, M. Kikuchi, and M. Asada. How can humanoid acquire lexicon?-active approach by attention and learning biases based on curiosity. *In: Proc. of IEEE International Conference on Intelligent Robots and Systems*, pages 3480–3485, 2006.
- [85] A. Ortony, G. Clore, and A. Collins. *The cognitive structures of emotions*. Cambridge University Press, 1988.

- [86] P.Y. Oudeyer. Developmental robotics. *Encyclopedia of the Sciences of Learning*, to appear in 2012.
- [87] P.Y. Oudeyer and F. Kaplan. Intelligent adaptive curiosity: a source of self-development. *In: Proc. of International Workshop on Epigenetic Robotics*, pages 127–130, 2004.
- [88] P.Y. Oudeyer and F. Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in Neurorobotics*, 1(6):257–262, 2007.
- [89] P.Y. Oudeyer, F. Kaplan, V.V. Hafner, and A. Whyte. The playground experiment: Task-independent development of a curious robot. *In: Proc. of the AAAI Spring Symposium on Developmental Robotics*, pages 42–47, 2005.
- [90] S. Pang, S. Ozawa, and N. Kasabov. Curiosity driven incremental lda agent active learning. *In: Proc. of International Joint Conference on Neural Networks*, pages 2401–2408, 2009.
- [91] L. Pape, C.M. Oddo, M. Controzzi, C. Cipriani, A. FérřÖster, M.C. Carrozza, and J. Schmidhuber. Learning tactile skills through curious exploration. *Frontiers in Neurorobotics*, 6:6, 2012.
- [92] R. Picard. *Affective computing*. The MIT Press, 1997.
- [93] R.W. Picard. Affective computing: challenges. *International Journal of Human-Computer Studies*, 59(1):55–64, 2003.
- [94] Rajagopal Raghunathan and Michel Tuan Pham. All negative moods are not equal: Motivational influences of anxiety and sadness on decision making. *Organizational behavior and human decision processes*, 79(1):56–77, 1999.
- [95] S.D. Ramchurn, D. Huynh, and N.R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.
- [96] S.D. Ramchurn, D. Huynh, and N.R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

- [97] S. Reiss. *Who am I? The 16 basic desires that motivate our actions and define our personalities*. The Berkley Publishing Group, 2000.
- [98] B. Renner. Curiosity about people: The development of a social curiosity measure in adults. *Journal of personality assessment*, 87(3):305–316, 2006.
- [99] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
- [100] E.T. Rolls. *Emotions in Humans and Artifacts*, chapter A Theory of Emotion, Its Functions, and Its Adaptive Value, pages 11–34. MIT Press, 2003.
- [101] H.J. Rong, Y.S. Ong, A.H. Tan, and Z. Zhu. A fast pruned-extreme learning machine for classification problem. *Neurocomputing*, 72(1):359–366, 2008.
- [102] S. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice hall Englewood Cliffs, 1995.
- [103] M. Salichs and M. Malfaz. A new approach to modeling emotions and their use on a decision making system for artificial agents. *IEEE Transaction on Affective Computing*, (99):56–68, 2011.
- [104] B.M. Sarwar, G. Karypis, J.A. Konstan, and J.T. Riedl. Application of dimensionality reduction in recommender system—a case study. *In Proc. of WEBKDD WORKSHOP*, 2000.
- [105] R. Saunders. *Curious design agents and artificial creativity: A synthetic approach to the study of creative behaviour*. PhD thesis, University of Sydney, 2002.
- [106] R. Saunders. Towards a computational model of creative societies using curious design agent. *In: Proc. of International Conference on Engineering Societies in the Agents World VII*, pages 340–353, 2007.
- [107] R. Saunders. Artificial creative systems and the evolution of language. *In: Proc. of the International Conference on Computational Creativity*, pages 36–41, 2011.

- [108] R. Saunders and J.S. Gero. A curious design agent. *In: Proc. of Conference on Computer Aided Architectural Design Research in Asia*, pages 345–350, 2001.
- [109] R. Saunders and J.S. Gero. Curious agents and situated design evaluations. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18(2):153 – 161, 2004.
- [110] R. Savitha, S. Suresh, and H.J. Kim. A meta-cognitive learning algorithm for an extreme learning machine classifier. *Cognitive Computation*, 6(2):253–263, 2014.
- [111] J. Schmidhuber. Adaptive confidence and adaptive curiosity. Technical Report FKI-149-91, Technische Universitat Munchen, April 1991.
- [112] J. Schmidhuber. Curious model-building control systems. *In: Proc. of IEEE International Joint Conference on Neural Networks*, pages 1458–1463, 1991.
- [113] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. *In: Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227, 1991.
- [114] J. Schmidhuber. Artificial curiosity based on discovering novel algorithmic predictability through coevolution. *In: Proc. of the Congress on Evolutionary Computation*, pages 1612–1618, 1999.
- [115] J. Schmidhuber. Exploring the predictable. *Advances in Evolutionary Computing*, pages 579–612, 2002.
- [116] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.
- [117] J Schmidhuber. Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. *In Proc: International Conference on Discovery Science*, pages 26–28, 2007.
- [118] J. Schmidhuber. Formal theory of creativity, fun and intrinsic motivation. *IEEE Transaction on Autonomous Mental Development*, 2(3):230–247, 2009.

- [119] J. Schmidhuber. Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Journal of the Society of Instrument and Control Engineers*, 48(1):21–32, 2009.
- [120] J. Schmidhuber. Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4(313), 2013.
- [121] F.F. Schmitt and R. Lahroodi. The epistemic value of curiosity. *Educational Theory*, 58(2):125–148, 2008.
- [122] F.D. Schoorman, R.C. Mayer, and J.H. Davis. An integrative model of organizational trust: Past, present, and future. *Academy of Management review*, 32(2):344–354, 2007.
- [123] P.D. Scott and S. Markovitch. Learning novel domains through curiosity and conjecture. In: *Proc. of International Joint Conference on Artificial Intelligence*, pages 669–674, 1989.
- [124] P.D. Scott and S. Markovitch. Experience selection and problem choice in an exploratory learning system. *Machine Learning*, 12(1-3):49 – 67, 1993.
- [125] Paul J Silvia. Interest and the curious emotion. *Current Directions in Psychological Science*, 17(1):57–60, 2008.
- [126] S. Singh, A.G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In: *Proc. of International Conference on Neural Information Processing Systems*, 2004.
- [127] E. Soloway, M. Guzdial, and K.E. Hay. Learner-centered design: The challenge for HCI in the 21st century. *Interactions*, 1(2):36–48, 1994.
- [128] C.D. Spielberger and L.M. Starr. Curiosity and exploratory behavior. pages 221–243, 1994.

- [129] G. Stojanov, A. Kulakov, and D. Clauzel. On curiosity in intelligent robotic systems. *In: Proc. of AAAI Fall Symposium on Interaction and Emergent Phenomena in Societies of Agents, Arlington, Virginia*, pages 44–51, 2006.
- [130] J. Storck, S. Hochreiter, and J. Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. *In: Proc. of the Int. Conference on Artificial Neural Networks*, pages 159–164, 1995.
- [131] K. Subramanian, S. Suresh, and N. Sundararajan. A metacognitive neuro-fuzzy inference system (McFIS) for sequential classification problems. *IEEE Transaction on Fuzzy Systems*, 21(6):1080–1095, 2013.
- [132] S. Suresh, N. Sundararajan, and P. Saratchandran. Risk-sensitive loss functions for sparse multi-category classification problems. *Information Sciences*, 178(12):2621–2638, 2008.
- [133] M. Tanti and S. Kennedy-Clark. Curriculum, technology & transformation for an unknown future. pages 963–967, 2010.
- [134] J. Togelius and J. Schmidhuber. An experiment in automatic game design. *In: Proc. of IEEE Symposium On Computational Intelligence and Games*, pages 111–118, 2008.
- [135] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
- [136] E. Uğur, M.R. Dogar, M. Cakmak, and E. Sahin. Curiosity-driven learning of traversability affordance on a mobile robot. *In: Proc. of IEEE International Conference on Development and Learning*, pages 13–18, 2007.
- [137] Noreen M Webb. Peer interaction and learning in small groups. *International journal of Educational research*, 13(1):21–39, 1989.
- [138] R.W. White. Motivation reconsidered: The concept of competence. *Psychological Review*, 66(5):297–333, 1959.

- [139] J. Wiecha, R. Heyden, E. Sternthal, and M. Merialdi. Learning in a virtual world: experience with using second life for medical education. *Journal of Medical Internet Research*, 12(1):e1. 1–27, 2010.
- [140] Y. Wilks. *Close engagements with artificial companions: Key social, psychological, ethical and design issues*. John Benjamins Publishing Company, 2010.
- [141] A. Wilson, M. Burnett, L. Beckwith, O. Granatir, L. Casburn, C. Cook, M. Durham, and G. Rothermel. Harnessing curiosity to increase correctness in end-user programming. *In: Proc. of SIGCHI Conference on Human Factors in Computing Systems*, pages 305–312, 2003.
- [142] M. Wooldridge. Intelligent agents: The key concepts. *Springer*, pages 3–43, 2002.
- [143] D. Wu, C.G. Courtney, B.J. Lance, S.S. Narayanan, M.E. Dawson, K.S. Oie, and T.D. Parsons. Optimal arousal identification and classification for affective computing using physiological signals: Virtual reality stroop task. *IEEE Transaction on Affective Computing*, 1(2):109–118, 2010.
- [144] M. Wu. Collaborative filtering via ensembles of matrix factorizations. *In Proc. of KDD Cup and Workshop*, pages 43–47, 2007.
- [145] Q. Wu and C. Miao. Curiosity: From psychology to computation. *ACM Computing Surveys (CSUR)*, 46(2):18, 2013.
- [146] Q. Wu and C. Miao. Modeling curiosity-related emotions for virtual peer learners. *Computational Intelligence Magazine*, 8(2):50–62, 2013.
- [147] Q. Wu, C. Miao, and B. An. Modeling curiosity for virtual learning companions. *In Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2014.
- [148] Q. Wu, C. Miao, and Z. Shen. A curious learning companion in virtual learning environment. *In: Proc. of IEEE International Conference on Fuzzy Systems*, pages 1–8, 2012.

- [149] W.M. Wundt. *Grundzüge physiologischen psychologie*. W.Engelman, 1874.
- [150] J.X. Xu, W. Wang, J.C.H. Goh, and G. Lee. Internal model approach for gait modeling and classification. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 7688–7691, 2006.
- [151] K. Yehuda. Factorization meets the neighborhood: a multifaceted collaborative filtering model. *In Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.
- [152] C.W.T. Yeu, M.H. Lim, G.B. Huang, A. Agarwal, and Y.S. Ong. A new machine learning paradigm for terrain reconstruction. *IEEE Geoscience and Remote Sensing Letters*, 3(3):382–386, 2006.
- [153] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato. A survey of trust and reputation management systems in wireless communications. *Proceedings of the IEEE*, 98(10):1755–1772, 2010.
- [154] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85, 2004.

Appendices

Appendix A. Pre-test Questions

Please answer the following multiple-choice questions.	
1	Water enters a plant through its roots by _ . (1 point) A. Osmosis B. Diffusion C. Active transport D. I have not learnt about this
2	When there is a higher concentration of mineral salts in the soil than in the root hair cells, the mineral salts enters the root hairs by _ . (2 points) A. Osmosis B. Diffusion C. Active transport D. I have not learnt about this
3	When there is a lower concentration of mineral salts in the soil than in the root hair cells, the mineral salts enters the root hairs by _ . (2 points) A. Osmosis B. Diffusion C. Active transport D. I have not learnt about this
4	Osmosis requires _ . (1 point) A. Partially permeable membrane B. Transport protein C. APT(energy) D. I have not learnt about this
5	Active transport requires _ and _ . (2 points) A. Partially permeable membrane B. Transport protein C. APT(energy) D. I have not learnt about this

BIBLIOGRAPHY

6	<p>Osmosis occurs when _ . (1 point)</p> <p>A. There is a lower concentration of water molecules in the soil than in the root hair cells</p> <p>B. There is a higher concentration of water molecules in the soil than in the root hair cells</p> <p>C. There is a lower concentration of mineral salts in the soil than in the root hair cells</p> <p>D. There is a higher concentration of mineral salts in the soil than in the root hair cells</p> <p>D. I have not learnt about this</p>
---	---

Appendix B. Post-test Questions

Please answer the following multiple-choice questions.	
1	<p>_ happens when there is a higher concentration of water molecules in the soil than in the root hair cells. (1 point)</p> <p>A. Osmosis B. Diffusion C. Active transport D. I did not learn this knowledge point in the game</p>
2	<p>The mineral salts enter the root hairs by _ , when there is a higher concentration of mineral salts in the soil than in the root hair cells. (2 points)</p> <p>A. Osmosis B. Diffusion C. Active transport D. I did not learn this knowledge point in the game</p>
3	<p>When there is a lower concentration of mineral salts in the soil than in the root hair cells, the mineral salts enters the root hairs by _ . (2 points)</p> <p>A. Osmosis B. Diffusion C. Active transport D. I did not learn this knowledge point in the game</p>
4	<p>Osmosis requires _ . (1 point)</p> <p>A. Partially permeable membrane B. Transport protein C. APT(energy) D. I did not learn this knowledge point in the game</p>
5	<p>Water enters the root by _ . (1 point)</p> <p>A. Osmosis B. Diffusion C. Active transport D. I did not learn this knowledge point in the game</p>
6	<p>Active transport requires _ and _ . (2 points)</p> <p>A. Partially permeable membrane B. Transport protein C. APT(energy) D. I did not learn this knowledge point in the game</p>

Appendix C. Questionnaire

Please respond to the following statements indicating the strength of your agreement/disagreement with the statement. Remember that there is no right or wrong responses.					
1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree					
1. The companion led me to think more deeply to solve the puzzle.	1	2	3	4	5
2. The companion encouraged me to explore for answers.	1	2	3	4	5
3. The companion assisted me to improve my knowledge of the content.	1	2	3	4	5
4. The companion was intelligent.	1	2	3	4	5
5. The companion was collaborating with me.	1	2	3	4	5
6. The companion was also learning.	1	2	3	4	5
7. The companion was curious.	1	2	3	4	5
8. The companion was human-like.	1	2	3	4	5
9. The companion was inquisitive.	1	2	3	4	5
10. The companion's behavior was natural.	1	2	3	4	5
11. The companion made me feel more confident to complete the tasks.	1	2	3	4	5
12. The companion was motivating.	1	2	3	4	5
13. The companion helped me to concentrate on the tasks.	1	2	3	4	5
14. I am sure that I can understand the knowledge about transport systems.	1	2	3	4	5
15. I am certain that I can do an excellent job on the problems about the transport system in the plants.	1	2	3	4	5
16. I will be able to learn the materials for the transport system in the plants.	1	2	3	4	5
17. The tasks in the game were easy to complete.	1	2	3	4	5
18. I got frustrated very often during the game play.	1	2	3	4	5
19. The learning companion helped me a lot to complete the tasks.	1	2	3	4	5
20. It would be more difficult to complete the tasks without the collaboration with the learning companion.	1	2	3	4	5
21. I was attentive while doing the task.	1	2	3	4	5
22. I was interested while doing the task.	1	2	3	4	5
23. I prefer to have the companion than playing alone.	1	2	3	4	5

Appendix D. Interview Questions

Questions:

1. Think of a friend in your mind.
2. What kind of movies does he/she like? (Note down the answer A_1)
3. What kind of movies he/she does not like? (Note down the answer A_2)
4. If he/she gives a low rating to a movie that belongs to the type of A_1 , will you feel surprised and want to watch that movie? Why?
5. If he/she gives a high rating to a movie that belongs to the type of A_2 , will you feel surprised and want to watch that movie? Why?